PROBABILISTIC SPACE MAPS FOR SPEECH WITH APPLICATIONS

A Thesis
Presented to
The Academic Faculty

by

Kaustubh Kalgaonkar

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
December 2011

PROBABILISTIC SPACE MAPS FOR SPEECH WITH APPLICATIONS

Approved by:

Professor Mark A. Clements, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Chin-Hui Lee
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor David V. Anderson
School of Electrical and Computer Engineering
Georgia Institute of Technology

Professor Erik I. Verriest
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Thomas Plöetz
DFG Research Fellow
School of Interactive Computing
Georgia Institute of Technology

Date Approved: 3 August 2011

To Vimla, Priti and Roma ...

ACKNOWLEDGEMENTS

I am ending my tenure at Georgia Institute of Technology the way it began, on a hot summer afternoon ...

I met Dr. Anderson in Topeka, KS in July 2005. He was on his way to Telluride, CO. I am enormously grateful that he took time off from long drive to meet with a prospective graduate student. We had a long discussion about my research interests and graduate studies. It was after that meeting that Dr. Anderson introduced me to Dr. Clements. Thanks to Dr. Anderson's recommendation that Dr. Clements accepted me as his student.

I want to express immense gratitude to my advisor Dr. Clements. I owe him an immense debt for his excellent guidance, support and encouragement throughout my graduate studies at Georgia Tech. He encouraged me to pursue ideas and topics independently and guided me when I was stuck. Most of all he taught me how to present my research to the community. He always promoted independence without abandoning any of his students. He encouraged me to do internships and to network with researches in the industry.

I would also like to take this opportunity to thank Dr. Bhiksha Raj. I worked with Dr. Raj during my two summer internships at Mitsubishi Electric Research Laboratories (MERL). Dr. Raj has a unique way of looking at problems and arriving at the solutions, I was lucky to work with him and learn those skills. Every discussion with Dr. Raj expanded my research horizons. The two internships at MERL have provided me with a lifelong friend.

I would also like to thank Dr. Michael Seltzer and Dr. Alex Acero, of Microsoft Research. I immensely benefited from their research expertise and experiences when

I did my internships at Microsoft Research and I truly appreciate their support and kindness.

I want to specially thank Dr. Chin Hui-Lee and Dr. David Anderson for reading my thesis and providing me feedback to improve it. I would also like to thanks Dr. Erik Verriest and Dr. Thomas Plötz for serving as members of my thesis committee.

I also want to express many thanks to the staff of Center for Signal and Image Processing (CSIP): Catherine Gholson, Tammy Scott, Patricia (Pat) Dixon, and Lisa Gardner. It is their hard work and support that allows us to concentrate on research and forget all the red tape that goes along with it.

I have been fortunate to have the company of some great friends. I will miss our Tuesday and Thursday lunch rituals. I will miss the occasional afternoon lunches at the Vortex. It has been a great pleasure to have Aditya Joshi, Amol Borkar, Sourabh Khire, Vikaram Appaia, Devangi Parikh, Ryan Palkki, Brett Matthews and Jonathan as friends. I am definitely going to miss long discussions (about anything under the sun) with Aditya and Ryan.

I want to express my deepest gratitude to my family for their unwavering support through out my long journey, I cannot possibly thank them enough. I want to thank my grandmother Vimla Kelkar, I am sorry you could not be here to see me graduate. I want to thank my parents Priti and Prakash Kalgaonkar, and my brother Ketan Kalgaonkar for their unconditional love and encouragement. I also want to thank my aunt and uncle Kirti and Satish Barde, I owe them a debt of gratitude for their support and love during my graduate studies.

Finally, Roma, love of my life, thanks for sticking around 14 years for me to find my first real job. Thanks for supporting me though some of the most difficult times of my life. Thank you for your unconditional love and unwavering support, without which this thesis and my Ph.D. would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of the proposed research is to develop a probabilistic model of speech production that exploits the multiplicity of mapping between the vocal tract area functions (VTAF) and speech spectra. Two thrusts are developed. In the first, a latent variable model that captures uncertainty in estimating the VTAF from speech data is investigated. The latent variable model uses this uncertainty to generate many-to-one mapping between observations of the VTAF and speech spectra. The second uses the probabilistic model of speech production to improve the performance of traditional speech algorithms, such as enhancement, acoustic model adaptation, etc.

Traditional mapping (Ψ) between two variables/subspaces can either be linear or nonlinear. In signal processing we often encounter mappings of the form $q = \Psi(p)$, where Ψ is a deterministic mapping function. In this thesis, we take a Bayesian approach towards the mapping function (Ψ). The mapping Ψ between variables/subspaces is extracted using a framework called probabilistic maps (PSMAPs). The PSMAPs have two components: latent variables to represent the subspaces containing observations $\mathbf{p} \in \mathcal{P}$ and $\mathbf{q} \in \mathcal{Q}$ as probability spaces, and a belief matrix that captures the information about the transform Ψ. The PSMAPs have two distinct advantages over the standard models:

1. No prior knowledge about the mapping Ψ is necessary. The mapping can be inferred from data.

2. Rich theory and principles of statistical estimation can be used to invert the mapping, even when it is nonlinear, complex or, unknown.

In this thesis, we propose to model the process of speech production with a probability map. This proposed model treats speech production as a probabilistic process with many-to-one mapping between VTAF and speech spectra. The thesis not only outlines a statistical framework to generate and train these probabilistic models from speech, but also demonstrates its power and flexibility with such applications as enhancing speech from both perceptual and recognition perspectives.

CHAPTER I

INTRODUCTION

Human speech is a unique signal. Speech is a result of several physiological pieces working together to convey a message. The main components of speech production are the lungs, glottis, and the vocal tract. The lungs and glottis together make the source that generates energy, which is shaped by a vocal tract filter to produce speech.

The vocal tract constrains the set of sequences of sounds that can be produced by a subject. The vocal tract is a mechanical configuration with components that include articulators such as lips, jaw, tongue, palate, and velum, in addition to various bones and soft-tissue. Each component has its own physical characteristics such as resonances, mass, inertia, etc., these characteristics impose restrictions on the configurations that a vocal tract can assume thereby, restricting the sequences that a subject can utter.

In addition to the physical constraints, articulator movements over time give rise to coarticulation. Coarticulation occurs due to the context of sounds that are being produced, or more accurately, it occurs because articulators must move smoothly from one orientation to another.

Speech production has fascinated scientists for centuries. Long before digital signal processing was invented, there were those who tried to build machines to create human speech. Some early legends of the existence of "speaking heads" involved Gerbert of Aurillac (1003 AD), Albertus Magnus (1198 – 1280), and Roger Bacon (1214 – 1294). In 1779, the Danish scientist Christian Kratzenstein, built models of the human vocal tract that could produce the five long vowel sounds.

In the past few decades, though, efforts were made to apply principles of physics

and fluid dynamics to model the vocal tract and the process of speech production [29, 78]. A variety of mathematical models for the vocal tract based on the equations suggested by Flanagan [29] have been proposed in the literature e.g., Sondhi and Schroeter [77]. These mathematical models have a large number of free parameters. Some of the parameters can be adjusted heuristically, others must be estimated using either pressure or volume velocity measurements at the lips.

With the advent of modern sensors, researchers have proposed the use of the articulator/gesture data to model of speech production [57, 43]. Models such as the Haskins CASY matches midsagittal vocal tracts to actual magnetic resonance imaging (MRI) data, and uses MRI data to construct a 3D model of the vocal tract. The availability of databases such as MOCHA [86], which consists of a set of real articulatory measurements and corresponding audio data has presented new avenues into the research of speech production models.

Speech synthesizers have benefited the most for the advances in speech production models. Recently the automatic speech recognition (ASR) community has turned its attention towards the use of articulatory information to augment the acoustic models trained from phonetic features. Studies such as [30] have shown that articulatory features will aid and improve the performance ASR systems. Frankel and colleagues [31, 30] demonstrated that ASR systems show improved accuracy when true articulatory features are used in conjunction with traditional features (such as MFCCs), but ASR systems do not show accuracy improvement if the traditional features are augmented with articulatory features estimated using speech inversion techniques on acoustic data.

The articulatory data has been used in speech synthesizers and ASR systems with varying degree of success. The performance of respective systems improves given the articulatory data is obtained from non-acoustic sources such as magnetic resonance imaging (MRI), electromagnetic articulograph (EMA), electroglottograph

(EGG), etectropalatograph (EPG), etc. The speech synthesis and ASR systems however do not benefit from articulatory information retrieved from acoustic signal using speech inversion [3, 53, 63, 64, 70, 75].

Speech inversion is a one-to-many process, because given spectrum of speech can generated by multiple articulator configurations. Speech inversion algorithms make assumptions to simplify the inversion process, and as a result the mapping between speech spectra and VT configuration ends up being one-to-one. Therefore, incorporating the articulatory parameters derived using the simplified models (speech inversion) seldom yield any benefit to the tasks such as ASR.

In this thesis we present a novel approach to model speech production. The new model is motivated by the fact that the relationship between articulators and speech is many-to-one, and we do not intend to simplify it. In this thesis we do not explicitly model the articulators. The articulators that include the vocal tract and glottal source are treated as latent variables, which are indirectly observed through parameters such as vocal tract resonances, line spectral pairs, etc. Application of the probabilistic model of speech production should be beneficial to traditional algorithms of speech enhancement, ASR, etc., because this model neither explicitly inverts speech signal, nor estimates the articulatory parameter.

In this thesis we will also present application of the probabilistic model of speech production to tasks of speech enhancement. We treat speech enhancement as two independent problems, (1) speech enhancement for perceptual quality improvement, and (2) speech enhancement for ASR.

1.1 Organization of Thesis

As stated in the title, the thesis proposes a probabilistic model for speech production. This model of speech production is applied to speech enhancement tasks such as, artificial bandwidth expansion (ABE), noise suppression and, acoustic model adaptation.

The thesis is organized as follows:

Chapter 2 discusses the uncertainty in speech inversion process. This chapter also examines the equivalence between uniform lossless tube model [58] and the inverse filter model [58]. The uncertainly in the speech inversion process is used to define a probabilistic model of speech production. This chapter further details the implementation of the probabilistic model of speech production using graphical models called probabilistic space maps (PSMAPs). Strategies for learning the parameters of PSMAPs followed by inference on the PSMAP complete this chapter.

Chapter 3 presents the first application of the PSMAP-based speech production model to artificial bandwidth expansion (ABE). The performance of proposed ABE algorithm is evaluated using a battery of subjective and objective tests. The chapter closes with the comparison of the performance of proposed scheme with the state-of-the-art ABE system.

Chapter 4 can be divided into two sections. First section presents an improvement to the probabilistic model of speech production. The new PSMAP models the temporal dynamics of the vocal tract by using a hidden Markov model to represent the VT area function subspace. This chapter also presents strategies to train the new constraint probabilistic space maps (CPSMAP). An algorithm for inference on the new model is also presented in this chapter. Second section of this chapter deals with the application of CPSMAP to task of noise suppression. The chapter closes with the subjective and objective evaluation of the CPSMAP-based speech enhancement system.

Chapter 5 presents a PSMAP-based algorithm for noise robust acoustic model adaptation. The performance of PSMAP-based ASR is evaluated and compared to existing acoustic model adaptation algorithms.

Chapter 6 discusses the future research direction of PSMAPs and closes with a summary of our contributions.

CHAPTER II

SPEECH PRODUCTION AND PSMAP

2.1 Model of Speech Production

The process of speech production can be modeled using two distinct blocks: (1) the glottal source and (2) the vocal tract transfer function (VTTF). Equation (1) mathematically represents the speech production process:

$$U_L(z) = G\Big(\frac{1}{D(z)}\Big)U_G(z), \tag{1}$$

where U_L is the volume velocity at the lips, U_G is the glottal source that drives the vocal tract (VT) filter $\frac{1}{D(z)}$ and G is the system gain.

To understand and model speech production one has to estimate $D(z)$ and $U_G(z)$. The VT estimation problem appears to be similar to system identification problem; the only difference between the two is that in the system identification problem we usually have access to the source (glottis and lungs). In a VT estimation problem we can neither separate VTAF from the source, nor independently control the source. Our inability to separate the VT response from the source makes VT response estimation an impossible problem given only the speech signal. The most successful methods of isolating the glottal source often use auxiliary sensors. Attempts have been made to separate the glottal source from the filter by measuring the volume velocity at the lips using p-mics [73], or by estimating the instants of glottal closure using EGG sensors around the neck of the talker [72]. Techniques that involve use of auxiliary sensors such as EGG, EPG, EMA, etc., are invasive in nature, since the sensor must be mounted on the subject. This kind of multimodal data collection is expensive and may not be available all the time. The standard databases and devices used for speech recognition, speech enhancement, etc., almost never have EGG or

volume velocity measurements.

Approximate VT measurements can be obtained from speech data alone. These approximate algorithms compensate for lack of glottal information with intelligent assumption about the losses in the system [83, 4]. In this section we will review two different methods of estimating VTAF and demonstrate how the inability to separate the glottal source from the VT-response leads to a probabilistic models of speech production.

2.1.1 Uniform Lossless Tube Model

The lossless tube model approximates the VT with M rigid lossless tubes of length 'l' and cross sectional areas $[S_1, S_2, \ldots, S_M]$. The VTTF $D(z)$ for a lossless tube model is computed by solving the partial differential equations for volume velocity and pressure. According to Markel [58], transfer function of the lossless tube model is given by Equation (2):

$$\frac{U_L(z)}{U_G(z)} = \frac{0.5(1+r_G)(1+r_{Lip})z^{-\frac{M}{2}} \prod_{m=1}^{M-1}(1+r_m)}{\begin{bmatrix} 1 & r_G \end{bmatrix} \left\{ \prod_{m=1}^{M-1} \begin{bmatrix} 1 & r_m \\ r_m z^{-1} & z^{-1} \end{bmatrix} \right\} \begin{bmatrix} 1 \\ r_{Lip} z^{-1} \end{bmatrix}}, \tag{2}$$

where r_m (Equation (3)) is the reflection coefficient at the boundary of tubes m and $(m+1)$:

$$r_m = \frac{S_{m+1} - S_m}{S_{m+1} + S_m}, \tag{3}$$

where r_G, r_{Lips} are the reflection coefficients at glottis and the lips respectively, S_m is the cross-section area of tube m, and $|r_m| \leqslant 1 \; \forall \; m$. To estimate the VTTF we must estimates the areas of all the tubes, including the areas of lip and glottal openings.

2.1.2 Inverse Filter Model

The VTTF can also be estimated using the inverse filter model of speech production. According to the inverse filter model, speech production is an autoregressive process

modeled by all-pole filter. The transfer function of the all-pole autoregressive filter is given by Equation (4):

$$H(z) = \frac{S(z)}{U_G(z)} = \frac{\widetilde{G}}{A(z)}, \tag{4}$$

where $\widetilde{G}$ is the gain, U_G is the driving glottal volume velocity, and $S(z)$ is the speech data. The lattice shown in Figure 1 represents the all-pole VT filter. The transfer function of all-pole VT (given by the Equation (5)) can be easily calculated by solving the lattice.

$$A(z) = \begin{bmatrix} 1 & k_g \end{bmatrix} \left\{ \prod_{m=1}^{M-1} \begin{bmatrix} 1 & k_m \\ k_m z^{-1} & z^{-1} \end{bmatrix} \right\} \begin{bmatrix} 1 \\ k_M z^{-1} \end{bmatrix}, \tag{5}$$

where k's are called the partial correlation coefficients or PARCORs. The PARCORs are estimated using the forward ($e[n]$) and the backward ($b[n]$) prediction errors. The Levinson-Durbin recursion [58] provides a very elegant solution to the PARCOR estimation problem.

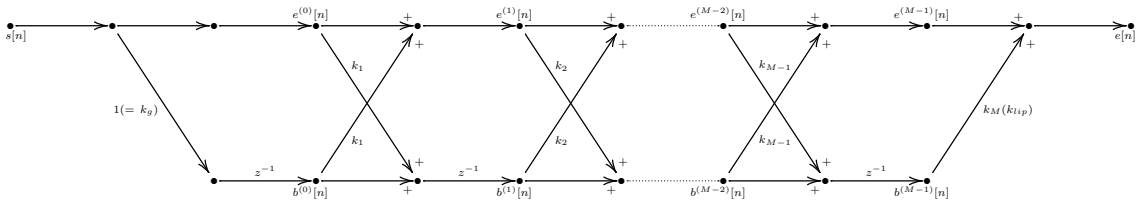


Figure 1: Lattice representation of the VT.

The '$m+1$' PARCOR is computed from the prediction errors of the m^{th} lattice, and is given by Equation (6):

$$k_{m+1} = \frac{-\sum_{n=-\infty}^{\infty} e^{(m)}[n] b^{(m)}[n]}{\sqrt{\sum_{n=-\infty}^{\infty} \left(e^{(m)}[n]\right)^2 \sum_{n=-\infty}^{\infty} \left(b^{(m)}[n]\right)^2}}, \tag{6}$$

where $e^{(m)}[n]$ and $b^{(m)}[n]$ are the m^{th} order forward and backward prediction errors.

The recursion to solve the lattice (Figure 1/ Equation (5)) is started with $e^{(0)}[n] = s[n]$ and $b^{(0)}[n] = s[n-1]$ ($k_G = 1$). It is interesting to note that although the set of k's uniquely define $A(z)$ in closed form, no such inversion exists. A unique inverse does exist if either k_m or k_G is known. In most cases, however, k_m or k_G cannot be estimated from acoustic data alone. Such an inverse would only exist in closed-form if either k_g or k_M were set to ± 1. These zero-loss coefficients values of reflection coefficients are unrealistic.

2.1.3 Relation between Lossless Tube and Lattice Models

Equations (1) and (4) suggest strong congruence between the vocal tract models obtained from inverse filtering (lattice method) and the lossless tube. Comparing the equations for $A(z)$ and $D(z)$, we can state that the two models of VTTF are equivalent. Under the assumption of equivalence of the two models, the PARCORs and reflection coefficients ($k_m = r_m$) are one and the same. The PARCORs and reflection coefficient equivalence essentially means that under zero-loss reflection coefficient assumption we can estimate VTAF using PARCORs.

PARCORs are estimated by making assumptions about the location of losses in the system ($k_{Lips} = 1$ is one such assumption). The assumption about the location of the loss is generally made to simplify the solution of the lattice recursion. The most common choice, is to associate all of the loss at one of the ends, either glottis ($r_{Lips} = k_{Lips} = 1$) [83] or lips ($k_g = r_g = 1$) [4]. The bundling of losses at one of the ends of the VT is unrealistic. Kalgaonkar and Clements [47] proposed a pragmatic approach to the solution of the lattice by distributing the losses across the glottal and the lip ends. The loss distribution was attained by imposing constraints on the VT, while solving for the parameters of the lattice. To solve the lattice for PARCORs [47], we imposed two smoothness constraints: (1) the difference in areas of adjacent tubes of a VT for a given frame should be minimum, and (2) the difference in the

VTAF of adjacent frame should be minimum.

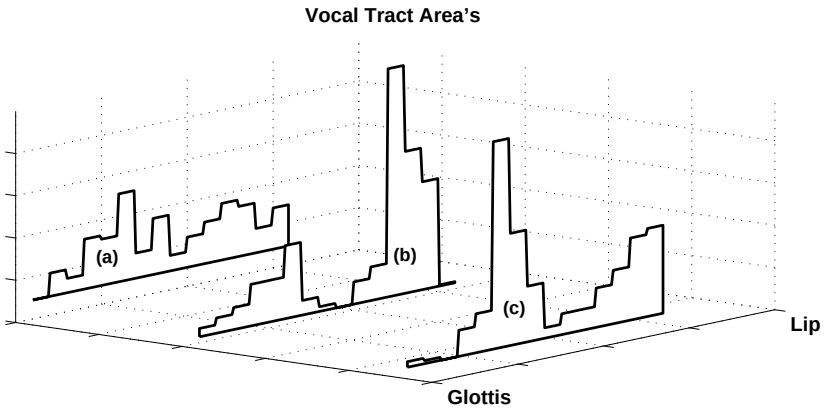


Figure 2: Area functions for a frame of speech with (a) $k_{lip} = 1$ all the loss at glottis. (b) $k_g \neq 1$ & $k_{lip} \neq 1$ loss at both lips and glottis. (c) $k_g = 1$ all the loss at the lips [47].

As an example Figure 2 shows three VT orientations producing the same impulse response ($A(z)$). The VTAF profile shown in Figure 2(a) was generated by assuming all the loss in the system is located at the glottis, and the VTAF profile Figure 2(c) was generated assuming all the loss in the system is at the lips. The VTAF profiles 2(a) and 2(c) are the extremes of solutions space, and there exist many other valid configurations of VT between the two extremes. Figure 2(b) is one such orientation of the vocal tract, where $k_g \neq 1$ and $k_{Lips} \neq 1$. The VT shown in Figure 2(b) was estimated using the procedure suggested by Kalgaonkar and Clements in [47].

The discussion thus far suggests that the relationship between speech spectra $A(z)$ or $\|S[j\omega]\|$ and VTAF is multivalued.[1] In theory it is possible to suggest distinct combinations of $|k_i| \leqslant 1 \quad \forall \quad i$, that produce lattices with the same impulse response $A(z)$ or $D(z)$. Each of these combinations of k's will however produce a distinct VT area-function.

This many-to-one mapping between VTAF and speech spectra is not exploited

[1]This many-to-one relationship between area functions and resonances of the VT was rigorously discussed in [76, 47].

in many of the current speech production models (e.g., [76]). The existing models ignore the many-to-one mapping in favor of simplicity. As we will see in this thesis, benefits of incorporating this mapping are immense when it comes to traditional speech applications such as enhancement, automatic speech recognition (ASR), etc.

The main impediment in acceptance of the many-to-one mapping has been the lack of a framework that is simple to learn and easy to apply to tasks such as ASR. In the next section we present a statistical model that can be used to learn multivalued mapping between VTAF and speech spectra. This new model of speech production can be easily trained from speech data alone.

2.1.4 PSMAPs as a Model of Speech Production

A PSMAP is a graphical model that uses a Bayesian approach to extract a functional mapping between two subspaces. Figure 3 shows a PSMAP that represents probabilistic mapping of VTAF to speech spectra. In this model, subspaces to be mapped are represented by number of distinct latent states (ρ, γ). Each latent state represents a distinct probabilistic basis for the subspace. An individual state from a Subspace $\mathcal{P}$ can map to multiple hidden states of Subspace $\mathcal{Q}$. The information about the mapping is captured in a discrete belief matrix $\boldsymbol{A}$, where $(\mathbf{p}, \mathbf{q})$ are the observation of the hidden states $(\boldsymbol{\rho}, \boldsymbol{\gamma})$ respectively.

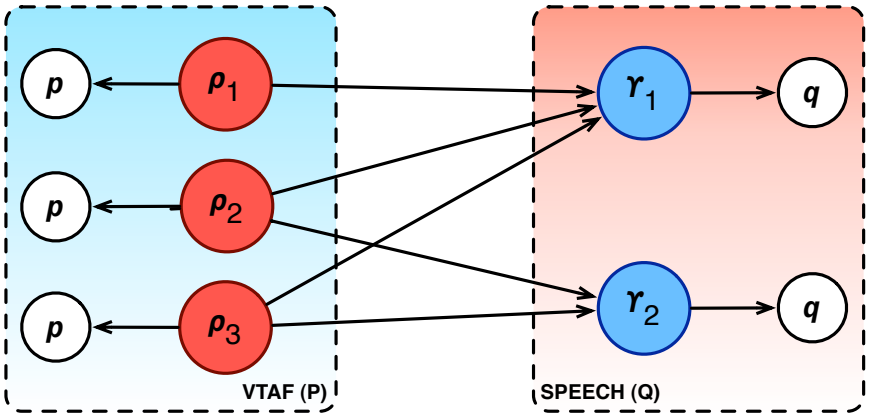


Figure 3: Graphical model for a PSMAP between two subspaces.

Discussions in Sections 2.1.1, 2.1.2 and 2.1.3 suggest that production of given spectra/quanta of speech is not defined by a unique source-filter pair. On the contrary, one can demonstrate that the right combination of source excitation and VTAF can produce any quanta of speech. Mathematically this can be written as $s(n) = \Psi(\text{VTAF})$, where $s(n)$ is a quantum of speech, and $\Psi()$ is the function that selects the appropriate VTAF and source pair. The uncertainty in estimating the true VTAF makes the function Ψ multivalued. We therefore propose that speech production is a probabilistic mapping between subspaces defined by the true VTAF and the speech spectrum. Figure 3 shows a PSMAP for such a model.

In a PSMAP for speech production ρ_i, γ_j are the latent states that generate the instances of VTTF ($\mathbf{p}$) and speech spectra ($\mathbf{q}$) with the probability $p(\mathbf{p}|\rho_i)$ and $p(\mathbf{q}|\gamma_j)$ respectively. As multiple VT orientations can produce similar speech spectra, each state ρ of Subspace $\mathcal{P}$ maps to multiples states γ of Subspace $\mathcal{Q}$. This many-to-one mapping is captured using a transition matrix $\boldsymbol{A}$, where $\sum_j a_{ji} = 1$ and $a_{ji} = p(\gamma_j|\rho_i)$ is the probability that the quanta of speech belonging to the latent state γ_i were generated from the vocal tract represented by the hidden state ρ_i. To understand the physical significance of the probability $p(\gamma|\rho)$, consider for a given state γ_n, if probability $a_{ni} > a_{nk}$; then it is more likely that the spectra of a given frame was generated by a VT configuration represented by state ρ_i than by state ρ_k.

True configuration of VT is always hidden. Reliable estimates of VTAF can be made with the help of auxiliary sensors such as EGG, MRI, etc. PSMAP-based model for speech production was designed to specifically to work in absence of true VT information. A PSMAP models the vocal tract as a hidden variable. We never really know the true configuration of the VT that produces a given spectral resonance, however, we can make educated guesses about the state of VT using observations derived from speech data. The PSMAP-based model of speech production is completely driven by this premise. As an example, line spectral pairs, PARCORs, vocal tract resonances,

etc., can be used as valid observations of the true vocal tract configuration.

2.2 Probabilistic Space Maps

Figure 4 shows the graphical model that is used to represent probabilistic space mapping between subspaces $\mathcal{P}$ and $\mathcal{Q}$. ρ and γ are hidden states that model the subspaces $\mathcal{P}$ and $\mathcal{Q}$ respectively. The gray rings represent the observed variables $\mathbf{p}$ and $\mathbf{q}$. The subspaces $\mathcal{P}$ and $\mathcal{Q}$ are modeled with N and M distinct states/basis. Each state in Subspace $\mathcal{P}$ is modeled with N Gaussians $\mathcal{N}(\boldsymbol{\mu}_\rho^n, \boldsymbol{\sigma}_\rho^n)$, where $n = 1, 2, \ldots, N$ and the states of Subspace $\mathcal{Q}$ are modeled with M Gaussians $\mathcal{N}(\boldsymbol{\mu}_\gamma^m, \boldsymbol{\sigma}_\gamma^m)$ where $m = 1, 2, \ldots, M$ (here $\boldsymbol{\sigma}$ denotes the variance and not the standard deviation).

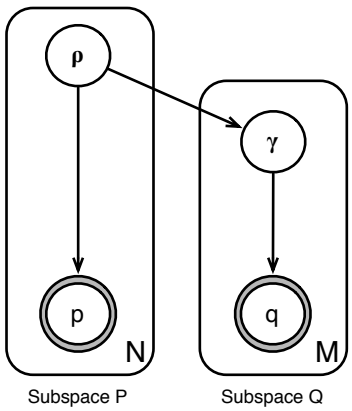


Figure 4: Graphical model representing the mapping between states of subspaces $\mathcal{P}$(VTAF) and $\mathcal{Q}$(Speech Spectra).

The relation between the states of $\mathcal{P}$ and $\mathcal{Q}$ is encoded in a transition matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$, where $\boldsymbol{A}(m, n) = a_{mn} = p(\gamma_m | \rho_n)$. The columns of transition matrix $\boldsymbol{A}$ sum to 1.

The parameters of the model can be estimated using expectation-maximization (EM) [18]. Before we can present the training algorithm for a PSMAP, it is important to understand (1) the impact the belief matrix $\boldsymbol{A}$ on the problem of inference using the graphical model, and (2) impact of the anatomical restrictions on mapping of

latent states (sparsity of the matrix $\boldsymbol{A}$)[2]. A good way to motivate the discussion about sparse models is to start with the problem of inference using a PSMAP.

2.2.1 Inference Using PSMAPs

The joint probability over the observed and latent variables $p(\mathbf{p}_t, \mathbf{q}_t, \rho_n, \gamma_m)$ can be written using the graphical model shown in Figure 4:

$$p(\mathbf{p}_t, \mathbf{q}_t, \rho_n, \gamma_m) = p(\mathbf{q}_t|\gamma_m)p(\gamma_m|\rho_n)p(\mathbf{p}_t|\rho_n)p(\rho_n), \tag{7}$$

where t is the time index.

Given a trained PSMAP $\mathcal{M} = \{\boldsymbol{P}, \boldsymbol{\Gamma}, \boldsymbol{A}\}$, and an observation $\mathbf{p}_t$, the minimum mean square error (MMSE) estimate of $\widetilde{\mathbf{q}}_t$ is given by Equation (8):

$$\widetilde{\mathbf{q}}_t = \mathbf{E}_{\mathbf{q}|\mathbf{p}}\{\mathbf{q}|\mathbf{p}\} = \int \mathbf{q} \cdot p(\mathbf{q}|\mathbf{p}) d\mathbf{q}. \tag{8}$$

The conditional probability $p(\mathbf{q}|\mathbf{p})$ required by the MMSE estimator can be expressed as the marginal of the joint probability and is given by Equation (7):

$$p(\mathbf{q}|\mathbf{p}) = \frac{p(\mathbf{p}\ \mathbf{q})}{p(\mathbf{p})} = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} p(\mathbf{p}, \mathbf{q}, \rho_n, \gamma_m)}{\sum_{n=1}^{N} p(\mathbf{p}|\rho_n)p(\rho_n)}. \tag{9}$$

Using Equations (7), (8), and (9), the MMSE estimate of $\widetilde{\mathbf{q}}_t$ can be written as

$$\widetilde{\mathbf{q}}_t = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N} \int \widehat{\mathbf{q}}\ p(\mathbf{q}|\gamma_m)p(\gamma_m|\rho_n)p(\mathbf{p}_t|\pi_n)p(\rho_n) d\widehat{\mathbf{q}}}{\sum_{n=1}^{N} p(\mathbf{p}|\rho_n)p(\rho_n)},$$

$$\widetilde{\mathbf{q}}_t = \sum_{m=1}^{M} \boldsymbol{\mu}_{\gamma}^{m} \left[\frac{\sum_{n=1}^{N} p(\gamma_m|\rho_n)p(\mathbf{p}_t|\rho_n)p(\rho_n)}{\sum_{n=1}^{N} p(\mathbf{p}|\rho_n)p(\rho_n)} \right], \tag{10}$$

$$\widetilde{\mathbf{q}}_t = \mathbb{M}_{\gamma}\boldsymbol{\nu}_t, \tag{11}$$

[2]A PSMAP without sparsity constraints on the belief matrix $\boldsymbol{A}$ can be estimated without any modification to the traditional EM algorithm. This PSMAP will be called the *simple models*.

where $\mathbb{M}_\gamma = [\boldsymbol{\mu}_\gamma^1 \vdots \boldsymbol{\mu}_\gamma^2 \vdots \cdots \vdots \boldsymbol{\mu}_\gamma^M]$ is the matrix of basis formed from the means of the Gaussian mapping Subspace $\mathcal{Q}$ and $\sum_m \nu_m = 1$ is the matrix of probabilities containing the belief for each basis. According to Equation (11), the estimate $\widetilde{\mathbf{q}}$ is the *convex sum* of the basis mapping the Subspace $\mathcal{Q}$.

2.2.2 Why Sparsity?

The problem with simple models is twofold. First, simple models require that the transition matrix $\boldsymbol{A}$ be a full matrix, which means all the states of Subspace $\mathcal{P}$ map to all the states of Subspace $\mathcal{Q}$, which is an unrealistic scenario. The vocal tract has finite flexibility that prevents arbitrary alignment of consecutive tubes. The rigidity of VT constrains the sounds that can be produced by a specific VTAF configuration, thereby resulting in a selective mapping between ρ and γ. A simple model does not provide the framework to explicitly impose the mapping constraint during training, and second, a small number of bases of subspace might prevent the capture of nuances in VT orientation. An overcomplete basis set is useful in resolving conflicts. Simply increasing the number of states in subspaces will lead to overfitting of the model. To elaborate and understand the significance of overcomplete basis, let us turn our attention towards MMSE estimates obtained using a PSMAP.

The MMSE estimates of $\mathbf{q}$ obtained from a PSMAP will always lie in the *convex hull* of the basis vectors of Subspace $\mathcal{Q}$. Any point outside the hull is estimated with error. The performance of the estimator depends on the number and the placement of these basis vectors. A PSMAP is trained by making an educated guess on the number of basis that would be required to map the subspaces $\mathcal{P}$ and $\mathcal{Q}$, where over-estimating the number of basis will lead to overfitting the model to available training data. The overfitted basis will fail to extract both the underlying mapping between the subspaces and structure of the data. The standard EM algorithm [19, 7] does not provide any means of preventing model overfitting, especially in cases were overcomplete

set of basis are used to model the subspaces. Sparse transition matrix $\boldsymbol{A}$ prevents the overfitting of data by eliminating unnecessary transitions between the states of subspaces $\mathcal{P}$ and $\mathcal{Q}$.

The differences between simple and sparse PSMAPs are discussed in detail in the Section 2.3.2.

2.3 Sparse EM for Training PSMAPs

The simple model maps every state ρ_n from Subspace $\mathcal{P}$ to every state γ_m in Subspace $\mathcal{Q}$ (with a probability of a_{mn}). In most of the cases, this might be completely unnecessary. A particular instance of $\mathbf{q}$ could be completely described by only a subset of the basis, implying each input state ρ_n only maps to a handful of the output states γ_m. This should result in a sparse transition matrix $\boldsymbol{A}$. For a PSMAP with overcomplete basis set, we might end up with a severely degenerated matrix $\boldsymbol{A}$ due to incorrect placement of basis in the subspaces. To stop the degeneration of the transition matrix, sparsity constraint on columns of matrix $\boldsymbol{A}$ must be imposed while training the PSMAP.

Various metrics have been applied to measure and impose sparsity. L_p norms are one of the most popular measures of sparsity [41].

In this study, however, we impose sparsity using an *entropic prior* [9]. The entropic constraints can be conveniently applied within the EM framework without many modifications to the existing algorithm.

Entropic prior $P_e(\theta)$ for a discrete probability distribution θ is:

$$P_e(\theta) \propto \exp\left(-\delta \mathcal{H}(\theta)\right), \tag{12}$$

where $\mathcal{H}(\theta)$ is the entropy of the discrete distribution θ, and the parameter δ controls sparsity. Positive values of parameter δ favor distributions with lower entropy.

A discrete Uniform distribution is the distribution with maximum entropy. A column of the transition matrix $\boldsymbol{A}$ is a discrete distribution. So if we minimize the

entropy $P_e(\theta)$ of the distribution that represents the columns of matrix $\boldsymbol{A}$, we force the columns of the matrix to be non-uniform. Jointly minimizing the entropy for all columns of the transition matrix $\boldsymbol{A}$ results in making the transition matrix sparse. This entropy constraint must be imposed during the M-step. We call this algorithm the sparse EM. Unlike the traditional EM, the M-step of sparse EM involves MAP estimation (where the prior for the MAP estimator is of the form given by Equation 12).

The goal of sparse EM is to make the columns of the matrix $\boldsymbol{A}$ and the prior on Subspace $\mathcal{P}$ – $p(\boldsymbol{\rho})$ sparse. Imposing sparsity forces the basis of subspaces to move in a direction that allows the model to provide better coverage of the subspaces.

2.3.1 Parameter Estimation

The EM algorithm consists of two steps:

1. E-step: Compute the *a posteriori* probability:

$$p(\rho_n, \gamma_m | \mathbf{p}_t, \mathbf{q}_t) = \frac{p(\mathbf{p}_t, \mathbf{q}_t, \rho_n, \gamma_m)}{\sum_{n=1}^{N} \sum_{m=1}^{M} p(\mathbf{p}_t, \mathbf{q}_t, \rho_n, \gamma_m).} \tag{13}$$

2. M-step: Maximize the complete data likelihood to estimate the parameters of the model:

$$\mathcal{L} = \mathbf{E}_{\gamma,\rho|\mathbf{pq},\mathcal{M}}\{\log p(\mathbf{p}_t, \mathbf{q}_t, \rho_n, \gamma_m)\}. \tag{14}$$

In order to impose sparsity, likelihood function $\mathcal{R}$ is generated by applying entropic constraints to the likelihood $\mathcal{L}$. The new augmented likelihood function $\mathcal{R}$ given by Equation (44):

$$\begin{aligned} \mathcal{R} &= \mathcal{L} + \tau\Big(\sum_n p(\rho_n) - 1\Big) + \delta \sum_n p(\rho_n) \log p(\rho_n) \\ &+ \sum_{n=1}^{N} \xi_n \Big(\sum_{m=1}^{M} p(\gamma_m|\rho_n) - 1\Big) \\ &+ \sum_{n=1}^{N} \epsilon \sum_{m=1}^{M} p(\gamma_m|\rho_n) \log p(\gamma_m|\rho_n), \end{aligned} \tag{15}$$

where τ and ξ_n are Lagrange multipliers that ensure $\sum_n p(\rho_n) = 1$ and $\sum_m a_{mn} = 1$.

The parameters δ and ϵ control the sparsity of the subspace $\mathcal{P}$ and the columns of transition matrix $\boldsymbol{A}$ respectively. The parameters δ and ϵ are tuned to obtain the desired sparsity.

The means ($\boldsymbol{\mu}$) and variances ($\boldsymbol{\sigma}$) of the Gaussian are estimated by maximizing the augmented likelihood $\mathcal{R}$ with respect to $\boldsymbol{\mu}_\rho, \boldsymbol{\sigma}_\rho, \boldsymbol{\mu}_\gamma$ and $\boldsymbol{\sigma}_\gamma$. The update equations for the means and variances given below:

$$\boldsymbol{\mu}_\rho^n = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n, \gamma_m|\mathbf{p}_t, \mathbf{q}_t)\ \mathbf{p}_t}{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n, \gamma_m|\mathbf{p}_t, \mathbf{q}_t)} \tag{16}$$

$$\boldsymbol{\sigma}_\rho^n = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n, \gamma_m|\mathbf{p}_t, \mathbf{q}_t)\ (\mathbf{p}_t - \boldsymbol{\mu}_\rho^n)^2}{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n, \gamma_m|\mathbf{p}_t, \mathbf{q}_t)} \tag{17}$$

$$\boldsymbol{\mu}_\gamma^m = \frac{\sum_{t=1}^{T}\sum_{n=1}^{N} p(\rho_n, \gamma_m|\mathbf{p}_t, \mathbf{q}_t)\ \mathbf{q}_t}{\sum_{t=1}^{T}\sum_{n=1}^{N} p(\rho_n, \gamma_m|\mathbf{p}_t, \mathbf{q}_t)} \tag{18}$$

$$\boldsymbol{\sigma}_\gamma^m = \frac{\sum_{t=1}^{T}\sum_{n=1}^{N} p(\rho_n, \gamma_m|\mathbf{p}_t, \mathbf{q}_t)\ (\mathbf{q}_t - \boldsymbol{\mu}_\gamma^m)^2}{\sum_{t=1}^{T}\sum_{n=1}^{N} p(\rho_n, \gamma_m|\mathbf{p}_t, \mathbf{q}_t).} \tag{19}$$

The entropic estimation of $p(\boldsymbol{\rho})$ and $\boldsymbol{A}$ is performed by maximizing the augmented likelihood $\mathcal{R}$. Unfortunately, a closed-form solution for these parameters does not exist. Maximizing Equation (15) with respect to $p(\rho_n)$ yields Equation (20) and maximizing Equation (15) with respect to $p(\gamma_m|\rho_n)$ leads to Equation (21):

$$\frac{\omega_n}{p(\rho_n)} + \delta + \delta \log p(\rho_n) + \tau = 0, \tag{20}$$

$$\frac{\Omega_{mn}}{p(\gamma_m|\rho_n)} + \epsilon + \epsilon \log p(\gamma_m|\rho_n) + \xi_n = 0, \tag{21}$$

where $\omega_n = \sum_t \sum_m p(\rho_n, \gamma_m | \mathbf{p}_t, \mathbf{q}_t)$ and $\Omega_{mn} = \sum_t p(\pi_n, \gamma_m | \mathbf{p}_t, \mathbf{q}_t)$ are the expected sufficient statistics. Equations (20) and (21) are similar in nature. Equations (20) and (21) form a system of simultaneous transcendental equations that can be solved[3] using the Lambert $\mathcal{W}$ function [15], to yield:

$$p(\rho_n) = \frac{-\omega_n/\delta}{\mathcal{W}(-\omega_n e^{1+\tau/\delta}/\delta)}, \tag{22}$$

Equations (20) and (22) form a pair for fixed-point iteration for τ/δ. We alternatively solve for Equations (20) and (22). The complete iterative procedure to estimate of $p(\boldsymbol{\rho})$ and matrix $\boldsymbol{A}$ is described in Algorithm 1.

Algorithm 1 MAP Estimation of $p(\boldsymbol{\rho})$

1: itt = 0
2: **for** itt < 5 or convergence **do**
3: Calculate $p(\boldsymbol{\rho})$ given (τ/δ) using (22).
4: Normalize $p(\boldsymbol{\rho})$ such that $\sum_n p(\rho_n) = 1$.
5: Compute (τ/δ) using (20), the current estimate of prior $p(\boldsymbol{\rho})$ and method suggested in Appendix A.
6: itt = itt + 1.
7: **end for**

The complete EM procedure to train a sparse PSMAP($\mathcal{M}$)[4] is given by equations (7), (13), (16), (17), (18), (19), (20), and (22).

2.3.2 Impact of Sparsity Constraints on the Model

In this subsection we analyze the impact of imposing sparsity on the performance of PSMAPs using a test example. This test problem has been specially created to highlight the benefits of sparse models over simple models.

The Subspace $\mathcal{Q}$ in this test problem consists of four distinct data clusters as shown in Figure 5. Cluster 2 is located such that it overlaps with Cluster 1 (y projection) and Cluster 4 (x projection). Taking the projections of the data in Subspace $\mathcal{Q}$ on

[3]Appendix A presents the detail of solving Equation (20) using Lambert $\mathcal{W}$ function.

[4]Appendix B explains the procedure for training PSMAPs with large number (> 100) latent states.

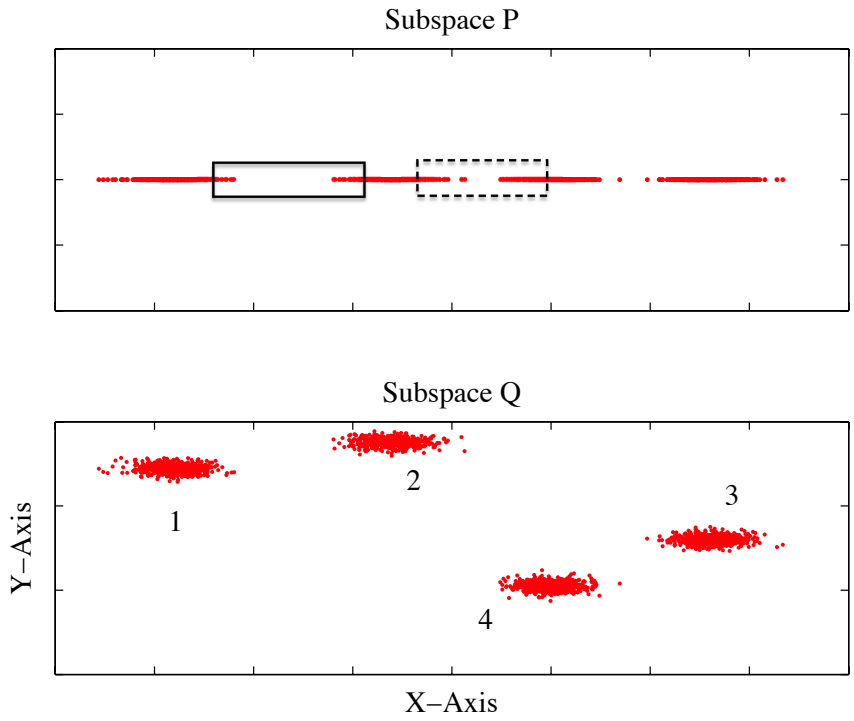


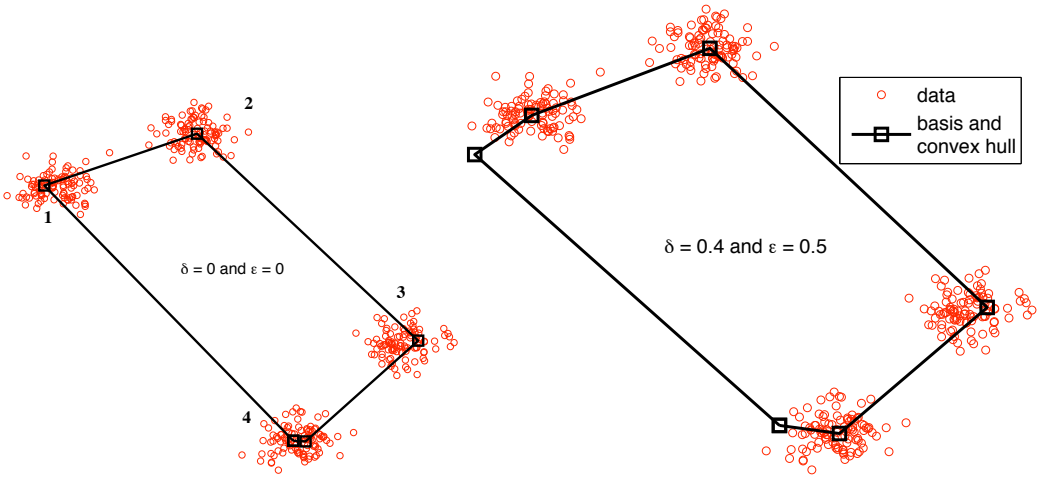
Figure 5: Test problem – Missing data estimation (Boxes in Subspace $\mathcal{P}$ indicate the region of ambiguity/overlap).

the x-axis results in data that forms Subspace $\mathcal{P}$. The boxed region in Subspace $\mathcal{P}$ indicates the region of ambiguity when moving from Subspace $\mathcal{P}$ to Subspace $\mathcal{Q}$. The x and y dimensions of the cluster (1 through 4) are uncorrelated. Most of the traditional algorithms will have difficulty in estimating $\widetilde{\mathbf{q}}$ given $\mathbf{p}$ because of the lack of correlation between the dimensions of Subspace $\mathcal{Q}$ and the overlap of the x-projection of data from Cluster 2 and Cluster 4.

Next we train two PSMAPs each with six bases in both the subspaces. Both sparse and simple PSMAPs were generated using the same training data. Figure 6(a) shows the basis trained using a simple PSMAP and Figure 6(b) shows the basis for the sparse PSMAP. The black squares indicate (□) the means/basis of the Gaussians mapping the Subspace $\mathcal{Q}$. The solid black line shows the *convex hull* formed by means of the Gaussians (basis).

The estimates of $\widetilde{\mathbf{q}}_t$ given $\mathbf{p}_t$ will always lie in the convex hull (Equation (11)) of

the basis. The simple model placed the Gaussians in a way that ended up over fitting the data by placing multiple Gaussians in Cluster 4. This placement resulted into a tighter convex hull and wasted computation. The estimates of $\widetilde{\mathbf{q}}$ using the simple model will have a problem reconstructing the missing data when $\mathbf{p}$ belongs to the overlap region.



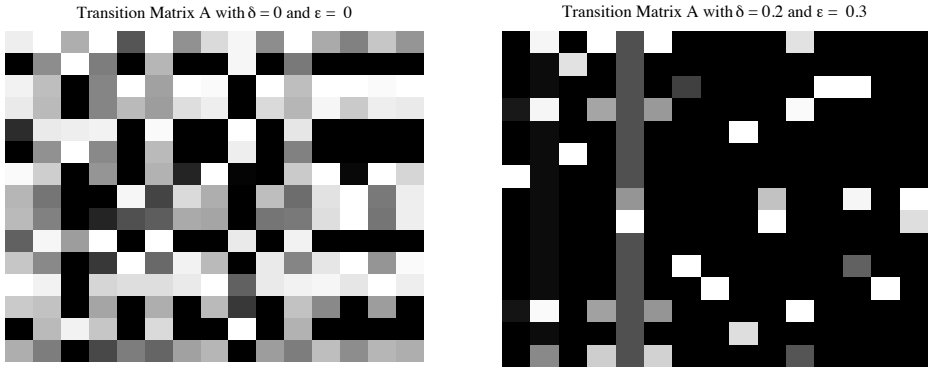
(a) Simple Model. (b) Sparse Model

Figure 6: Subspace $\mathcal{Q}$ data, basis, and the convex hull.

The second sets of basis were trained using a sparse PSMAP (Figure 6(b)). The parameters $\epsilon = 0.5$ and $\delta = 0.4$ were used to impose sparsity on $\boldsymbol{A}$ and $p(\boldsymbol{\rho})$ respectively. The sparse PSMAP tries to discover the structure within the data and ends up assigning more Gaussians to clusters 1 and 4. This intelligent placement of clusters increases the area of the convex hull, thereby allowing better coverage of Subspace $\mathcal{Q}$. We also observed that the sparse PSMAP have lower MSE while reconstructing the missing data.

Figure 7 shows section of two transition matrices $\boldsymbol{A}$. The matrices were generated for a bandwidth expansion problem. Both the simulations were carried out using the same data. The training of both simple and sparse PSMAPs was started with the same initial conditions. Most of the elements of the matrix $\boldsymbol{A}$ for a sparse PSMAP are zero; this zeroing of transition probabilities weeds out unnecessary dependencies

between states of subspaces reducing both computation and complexity. In most cases we observed that a sparse PSMAP would have around 50% non-zero entries in the transition matrix.



(a) No Sparsity imposed.

(b) With sparsity constraints $\delta = 0.2$ and $\epsilon = 0.3$.

Figure 7: Sparse vs. Simple transition matrix $\boldsymbol{A}$. (Darker values indicate $p(\gamma|\pi)$ closer to zero).

2.4 Conclusions

In this chapter we presented a statistical framework (PSMAP) to learn this multi-valued mapping between two datasets. PSMAPs were used to propose a new model for speech production. This PSMAP-based model of speech production captures the many-to-one mapping between the VTAF and speech spectrum. The strength of PSMAP is its ability to learn many-to-one mapping without the use of non-acoustic auxiliary sensors or explicit knowledge of true VT configurations.

In this chapter we presented algorithm to train PSMAP, we also presented a modified expectation-maximization algorithm to train a sparse variant of the PSMAP. Sparse PSMAPs are useful in cases where subspaces are modeled with a very large number of latent states. Sparse probabilistic space maps are better at extracting underlying relationships between the latent states, as a consequence they prevent overfitting of the model to training data. Sparse probabilistic maps are also computationally efficient for large models.

CHAPTER III

ARTIFICIAL BANDWIDTH EXPANSION

3.1 Introduction

Artificial bandwidth expansion (ABE) is a process of automated addition of missing high and low frequency components to a bandlimited speech signal. Listening tests have shown that the presence of high frequency components in speech make it perceptually more pleasing thereby improving its perceived quality [82] as measured in MOS [1] tests. For telephone speech ABE yields an average improvement of 1.3 MOS points. Intelligibility of meaningless syllables in a phone conversation is about 90% only, as a result, users sometimes have to spell out unfamiliar words, or words that are used out of context.

All the telephone communication is still limited to 8 kHz speech, this is due to the fact that new telecommunication technologies must interoperate with the old ones, which mostly operate at or below 8 kHz. Most of the ABE techniques focus on extending the bandwidth of a telephony (300 Hz to 3700 Hz) signal producing a speech signal in the range 0 Hz to 8000 Hz.

The goal of ABE is to improve perceptual quality of speech. To achieve this goal artifact-free synthesized speech is one of the pivotal requirements of a good ABE system.

Various techniques have been proposed for this task over the years. All existing ABE techniques can be classified into aliasing-based methods, statistical methods, or codebook-based methods. Aliasing-based methods (e.g., [87]) employ a nonlinear transformation to construct the absent high frequency components by aliasing low frequency components. Some methods, such as those suggested by Yoshida and

Chennoukh [88, 12] use codebooks to generate a map between the low and the high frequencies of the spectrum. This codebook is used to reconstruct the missing high frequencies using the low frequency information extracted from the narrowband speech. The performance of codebook-based ABE methods can be improved if linear combination of the high frequency spectrum is used to reconstruct missing frequencies (e.g., [5]).

Statistical methods, such as those proposed in [48, 40], model the relationship between the lower and upper band frequency components using Hidden Markov Models (HMM), Gaussian Mixture Models (GMM), etc. The trained statistical models are then used to estimate the missing frequency components.

In the next section we will present a PSMAP-based ABE scheme. This method does fall in to the category of statistical methods for ABE.

3.2 PSMAP for Artificial Bandwidth Expansion

Most of the existing ABE techniques are based on linear prediction (LPC) analysis-synthesis systems. These ABE systems must solve two problems: (1) estimate the broadband LPC, and (2) estimate the broadband LP residue.

There are two distinct problem associated with such a system. First, traditional techniques estimate broadband LPC from the narrowband LPC using a codebook or a statistical model. LPC coefficients are very sensitive to floating point errors. Small decimal errors in the values of LPC coefficient often result in large changes in the spectrum of the analysis filter, therefore, directly using codebooks for mapping broadband and narrowband LPC is not a very good proposition. Second, almost all the residue extension techniques use some form of aliasing/modulation-based schemes to extend the narrowband residue. These techniques exploit the harmonic nature of speech to extend the narrowband residue. The variance (LP gain) of the extended residue seldom matches the variance of the true broadband residue. This energy

mismatch in the broadband and narrowband residue results into discontinuity at the boundary (3700 Hz) of broadband and narrowband spectra. These spectral mismatch manifests as "audible glitches" in the synthesized audio.

A better approach for the ABE problem is to work in the spectral domain since human perception is relatively insensitive to time-invariant phase distortions. This quality of the human auditory systems allows us to focus our effort of bandwidth extension of the magnitude spectrum. As we will observe in the results section, spectral domain ABE system has better performance than the LPC-based ABE system. ABE in spectral domain involves extending the magnitude spectrum and the phase spectrum. ABE of the magnitude spectrum is accomplished using a PSMAP. Since the human perception is insensitive to phase distortion, the phase spectrum is extended using a simple linear transform $\mathbf{W}$.

The ABE system has three important stages (Figure 8): *broadband magnitude estimation*, *broadband phase estimation*, and *post processing*. The next subsection presents implementation details of the three stages.

3.2.1 Broadband Magnitude Estimation

The broadband magnitude spectrum is estimated using a PSMAP. The Subspace $\mathcal{P}$ is the space formed by the observations of the vocal tract. A PSMAP for bandwidth expansion uses LPC-MFCC of narrowband speech as the observations of the VT. The Subspace $\mathcal{Q}$ is formed using log-spectra of broadband speech. For the problem of ABE we observed that LPC-MFCC are better than LPC, line spectral pairs (LSP), and log area ratios (LAR) at capturing the VT variations.

Let $\mathbf{p} \in \mathbb{R}^k$ and $\mathbf{q} \in \mathbb{R}^l$ represent observations of Subspace $\mathcal{P}$ and Subspace $\mathcal{Q}$ respectively. Once trained, this PSMAP will statistically model the relationship between speech spectra and VT configurations responsible for its production. This PSMAP also indirectly captures the relationship between narrowband and broadband

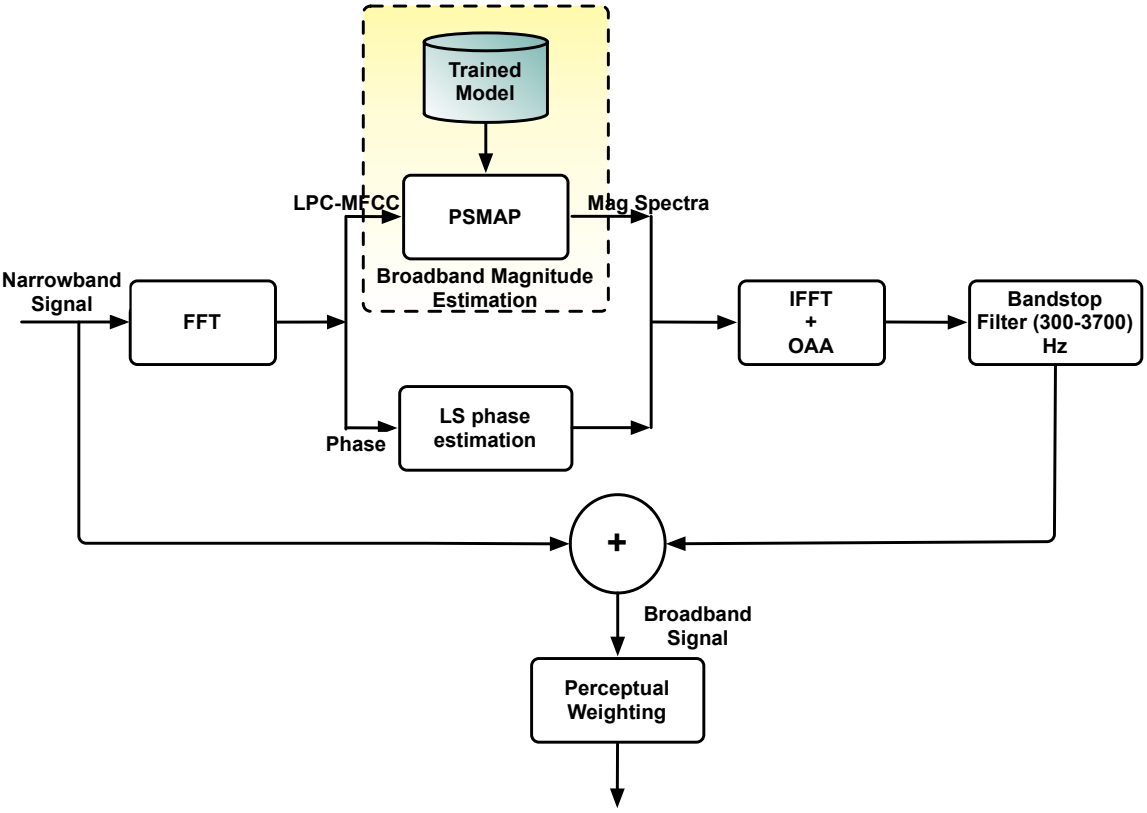


Figure 8: Block diagram of ABE system.

speech spectra.

The trained PSMAP will be used to estimate broadband magnitude spectrum from the narrowband LPC-MFCC observations. Broadband log spectrum for frame $\widetilde{\mathbf{q}}_t$ can be estimated from the narrowband LPC-MFCC coefficient $\mathbf{p}_t$ using the MMSE estimator given by Equation (10).

3.2.2 Broadband Phase Estimation

The phase of broadband speech $\widetilde{\phi}_q$ is estimated from the phase of the narrowband speech ϕ_q using a linear transform $\boldsymbol{W}$ (Equation (23)):

$$\widetilde{\phi}_q = \mathbf{W}\phi_{\mathbf{p}}. \tag{23}$$

The transform matrix $\boldsymbol{W}$ is learned from training data using a linear least squares

estimator given by Equation (24):

$$\boldsymbol{W} = \Phi_{\mathbf{q}}\Phi_{\mathbf{p}}^{\dagger}, \tag{24}$$

where $\Phi_{\mathbf{p}} = [\phi_{\mathbf{p},1}, \phi_{\mathbf{p},2}, \ldots, \phi_{\mathbf{p},T-1}, \phi_{\mathbf{p},T}]$, $\Phi_{\mathbf{q}} = [\phi_{\mathbf{q},1}, \phi_{\mathbf{q},2}, \ldots, \phi_{\mathbf{q},T-1}, \phi_{\mathbf{q},T}]$ are the matrices of phases of the narrowband and broadband speech respectively, and † is the pseudo-inverse operator.

3.2.3 Post Processing

The broadband magnitude and phase spectra are combined to recover the complex broadband spectrum. Broadband speech is synthesized from the broadband spectrum using the overlap-and-add (OAA) method [65]. The synthesized broadband speech is passed through a bandstop filter with cutoff of 300 Hz and 3700 Hz to retain only the frequencies missing from the narrowband signal. The filtered speech is combined with the original unprocessed utterance to synthesize complete broadband signal.

The reconstructed speech sometimes has a high-frequency hiss. The high-frequency hiss is a result of noise introduced during the broadband magnitude and phase estimation process. The high-frequency hiss can be suppressed by passing the synthesized speech through a perceptual filter. The perceptual filter is based on the LP-analysis filter and is given by:

$$W(z) = \frac{1 - A(z/\alpha)}{1 - A(z/\beta)} \qquad 0 < \beta < \alpha \leq 1, \tag{25}$$

where $A(z)$ is the linear prediction polynomial.

3.3 Experiments and Results

To evaluate the performance of PSMAP-based ABE system, experiments were performed on a subset of the Wall Street Journal database (WSJ) [66]. The data was split into training and test sets. The training set consisted of 5 – 6 min of speech data from each of the three male and three female subjects. The test data was split

into two sets. Set A included data from the six subjects in the training set. The test utterances in Set A were not included in the training set. Set B consisted of a male and a female speaker not included into the training set. The purpose of Set B is to evaluate the performance of PSMAP-based ABE system on unseen user data.

The WSJ utterances were recorded and sampled at 16 kHz. The original WSJ utterance was treated as broadband data. The narrowband data (sampled at 8 kHz) were generated by bandpass filtering the broadband data with an 8^{th} order Chebyshev filter with cutoffs of 300 Hz and 3700 Hz. The filtered data was downsampled by two to generate the narrowband speech.

A 25 ms Hamming window was used to analyze the speech. Adjacent frames had a 15 ms overlap. The utterances were mean normalized to remove any DC contribution before analysis. All the utterances were pre-emphasized before feature extraction.

Thirteen LPC-MFCCs were obtained in a standard way from 23 mel-filter banks applied to the LPC spectrum of the narrowband speech [89]. Regression delta coefficients ($\Delta\mathbf{p}$) were appended to the LPC-MFCC; this composite vectors ($[\mathbf{p}^T, \Delta\mathbf{p}^T]^T$) were used as features for the narrowband Subspace $\mathcal{P}$. The delta coefficients were computed using regression window of size 2, using Equation (26)

$$\Delta\mathbf{p}_t = \frac{\sum_{\theta=0}^{2} \theta\Big(\mathbf{p}_{t+\theta} - \mathbf{p}_{t-\theta}\Big)}{2\sum_{\theta=0}^{2} \theta^2}. \tag{26}$$

A 512 point DFT was performed on each broadband frame of speech; 257 bins of magnitude spectrum were retained to form the features of broadband Subspace $\mathcal{Q}$.

Evaluations of the ABE system were performed on both test sets A and B. Four different kinds of PSMAPs were generated to evaluate the performance of the algorithm under different conditions. Multiple tests were carried out using the following four models:

Model A: Six speaker-dependent models.

Model B: A speaker-independent model was generated using training data from all six (three male and three female) subjects.

Model C: A model for all male subjects was generated.

Model D: A model for all female subjects was generated.

Performance of all the models was evaluated using two metrics: perceptual evaluation of speech quality (PESQ)[44] and spectral distortion (SD). Equation (27) illustrates the procedure to compute spectral distortion.

$$D^2 = \frac{1}{f_s}\int_0^{f_s} \big(20\log_{10}(P_{ss}(f)) - 20\log_{10}(\widehat{P}_{ss}(f))\big)df, \tag{27}$$

where f_s is the sampling frequency in Hz and $P_{ss}(f) = (|A(\exp(\jmath 2\pi f/f_s))|)^{-1}$; $A(z)$ is the linear prediction polynomial.

3.3.1 Objective Test Results

Table 1 shows the PESQ and SD scores for speaker-dependent models (Models A) of various subspace sizes. We observed that the performance of PSMAP is order dependent, and improves with the size of the PSMAP. The best performance was obtained for a model of order $N = 193$ and $M = 161$ and the sparsity parameter $\epsilon = 0.4$ with $\delta = 0.1$. There is no closed-form method to determine the sparsity parameters in advance; choice of the parameters is based on multiple trials. Increasing the model size does result in improvement of both the objective scores. The PSMAP with the model size (161×193) demonstrates a relative improvement of 5% in PESQ[1] score and a relative improvement of 16.8% in the SD[2] over the smallest PSMAP of size (50×50).

[1]Relative improvement in PESQ scores is calculated by comparing absolute improvement to the failure rate: $\left(\frac{PESQ_1 - PESQ_2}{4.5 - PESQ_2}\right)$

[2]Relative improvement in SD scores is calculated by comparing absolute improvement to the failure rate: $\left(\frac{SD_1 - SD_2}{0 - SD_2}\right)$

Table 1: Performance of PSMAP-ABE for Set A (N - size of Subspace $\mathcal{P}$ and M - Size of Subspace $\mathcal{Q}$)

(N,M)		(161,193)		(100,120)		(70,90)		(70,70)		(50,50)	
	Subject	**PESQ**	**SD**	**PESQ**	**SD**	**PESQ**	**SD**	**PESQ**	**SD**	**PESQ**	**SD**
MALE	M-1	**4.2551**	3.4703	4.2550	3.5468	4.2500	3.6587	4.2467	3.8274	4.2450	4.1722
	M-2	**4.2896**	3.2945	4.2889	3.3405	4.2828	3.4297	4.2799	3.5358	4.2757	3.7399
	M-3	**4.1463**	3.1732	4.1391	3.2305	4.1320	3.3641	4.1319	3.4877	4.1264	3.5750
FEMALE	F-1	**4.2363**	3.5223	4.2335	3.5658	4.2317	3.7515	4.2293	3.8978	4.2211	4.2774
	F-2	**4.2735**	3.5062	4.2720	3.5646	4.2716	3.6027	4.2693	3.7188	4.2675	3.8233
	F-3	**4.2374**	3.3385	4.2344	3.4239	4.2305	3.6064	4.2265	3.8750	4.2209	4.1485

The performance of ABE system saturates once the model size reaches (161 x 193), any further increment in latent states does not result in improvement in performance.

The second sets of experiments were aimed at the comparison of broadband and narrowband speech. The PESQ standard tests [44] do not allow comparison of utterances sampled at different frequencies. To overcome this problem, bandpass filtered narrowband speech was not downsampled to 8 kHz. For these set of experiments, bandpass filtered speech was treated as narrowband speech and was used for PESQ comparisons. The PESQ scores were calculated for two cases:

1. Comparing ABE speech with the original broadband utterance.

2. Comparing narrowband speech with the original broadband utterance.

The ABE speech was generated using speaker-independent PSMAPs (Model B). Figure 9 presents the comparison of PESQ scores for ABE broadband speech and narrowband speech. ABE speech has higher PESQ scores for all subjects. We observed an average relative improvement[3] of 61.72% in the PESQ score of ABE broadband speech over the original narrowband speech. Based on these results, we can state that quality of ABE speech produced by using a PSMAP is better than the quality of unprocessed narrowband speech.

A speaker-independent PSMAP was used for ABE of utterances from unseen speakers (Set B). The utterances of unseen speakers show smaller improvement in PESQ scores. This can be attributed to the use of speaker-independent models for ABE of Set B. We observed an average relative improvement of 65.02% in PESQ scores for the utterances of speakers in Set A, and an average relative improvement of 51.8% in PESQ scores for utterances of speakers in Set B.

The next set of experiments presents a comparison between sparse and simple

[3]Average relative improvement in PESQ score is calculated by comparing absolute improvement to the failure rate: $\left(\frac{PESQ_{ABE}-PESQ_{nb}}{4.5-PESQ_{nb}}\right)$, where 'nb' is the PESQ score for narrowband speech.

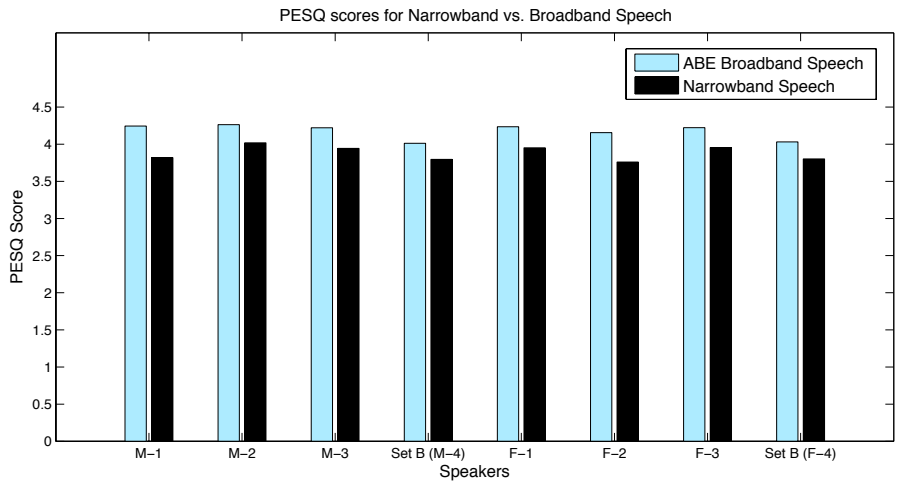


Figure 9: Comparison of PESQ scores for broadband and narrowband speech.

PSMAPs. Both models were trained using speech from all six users. Both models had the same number of states $(M, N) = (193, 161)$. The sparsity parameter ϵ was set to 0.4. Figure 10 shows the boxplots for PESQ scores of test data from both Set A and Set B. The performance (PESQ scores) of both the sparse and simple models is similar for Set A, however, the sparse model has better PESQ scores for unseen users (Set B, M-4 & F-4). We observed a relative improvement of 3% in PESQ score when sparse PSMAP was used for ABE of utterances in Set B. Using sparse PSMAP for ABE of utterances in Set A resulted in a relative improvement of 0.07% in PESQ score.

Imposing sparsity on the PSMAP forces some of the basis of the subspaces to occupy the low probability regions of the subspace. This movement of the basis prevents overfitting of the data and results in latent states that capture the underlying the structure of the VT producing the spectrum. This property of the sparse PSMAP allows it to model unseen user data better than the simple PSMAP. Therefore, when using sparse PSMAP we observe larger improvement on unseen user data. We have observed through experiments, that PSMAPs with small number of hidden states (< 32) do not benefit from sparse EM.

We also observed that sparse PSMAPs provide computational advantage over the

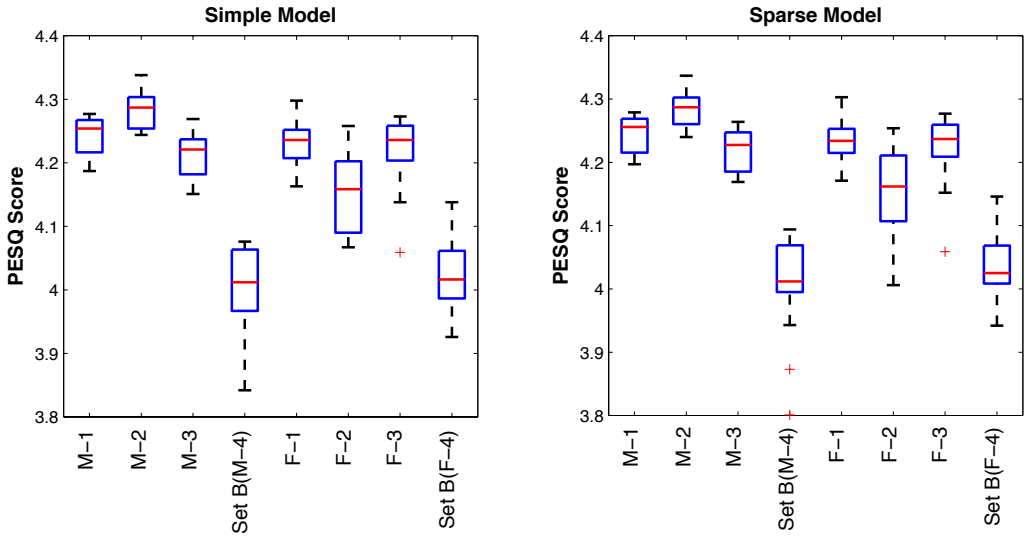


Figure 10: Comparison of PESQ scores for sparse vs. simple PSMAPs.

simple models. The transition matrix $\boldsymbol{A}$ generated by sparse EM contains only 40% non-zero entries.

The next set of experiments compare three different types of PSMAPs: speaker-dependent PSMAPs (Model A), gender-dependent PSMAPs (Model C/D), and speaker-independent PSMAPs (Model B). The results of the experiments are presented in Table 2. The results shown in Table 2 were produced using best in class models. The PSMAPs used to compute the SD had different sizes ($size(A) < size(C/D) < size(B)$). The size of the PSMAP depends on the variability in data that is being modeled; consequently speaker-independent models have the largest model size. The sizes of models A, C/D, and B were (161 $\times$ 193), (250 $\times$ 250) and (350 $\times$ 350) respectively.

The speaker-dependent PSMAP (Model A) generate the best VT model for a single user, thereby these PSMAPs provide the best reconstruction of the missing frequencies for a given speaker. A comparison between gender-dependent PSMAPs (Models C/D) and speaker-independent PSMAPs (Model B) highlights the fact that even with higher model order, speaker-independent models are not as successful as the gender-dependent models in reconstruction of missing frequencies.

Table 2: Model vs. SD performance

Subject	Model A	Model C/D	Model B
M-1	3.4703	3.5278	3.7193
M-2	3.2945	3.5094	3.5425
M-3	3.1732	3.4732	3.6511
F-1	3.5223	3.617	3.6936
F-2	3.5062	3.5785	3.6626
F-3	3.3385	3.6478	3.7196

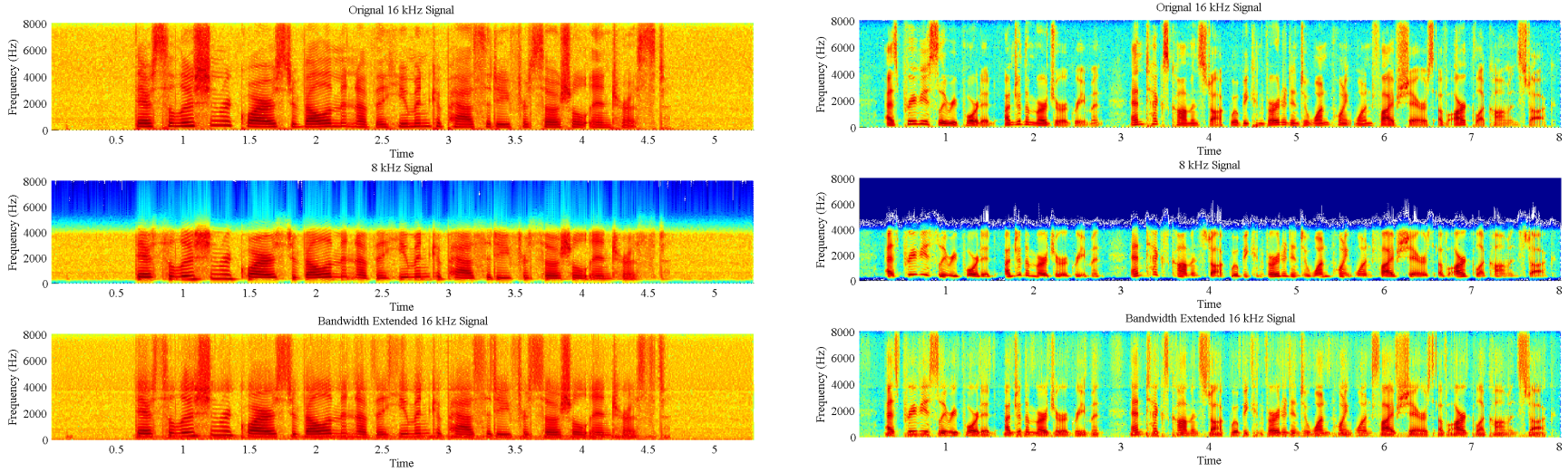
To roughly maintain the same objective quality with increasing variability in data, the gender-dependent PSMAPs (Model C/D) are roughly 1.7 times lager than the speaker-dependent PSMAPs (Model A), and the speaker-independent PSMAP (Model B) is roughly 3.3 times larger than the speaker-dependent PSMAP.

We can appreciate the importance of sparse EM when we compare the speaker-dependent PSMAP and the speaker-independent PSMAPs. A speaker-dependent model has roughly 31,000 elements in transition matrix $\boldsymbol{A}$ and, the speaker-independent PSMAP has around 122,500 elements in its transition matrix $\boldsymbol{A}$. Upon training a sparse-speaker-independent PSMAP, we observed that the transition matrix $\boldsymbol{A}$ only had 49,000 active entries. Hence the sparse PSMAP will have the same computational complexity as that of the speaker-dependent PSMAP. This improvement in computation is achieved without sacrificing quality.

Figure 11 shows spectrograms of original and reconstructed speech for a male and a female speaker. In both cases the algorithm is able to reconstruct the missing frequencies both in the 0 – 300 Hz and 3700 – 8000 Hz regions.

3.3.2 Subjective Test Results

Artifact free speech is one of the fundamental requirements of successful ABE algorithm. Objective tests do not measure or quantify the artifacts present in synthesized speech. Subjective tests are therefore necessary to evaluate the quality of synthesized speech. Subjective tests are plagued with various problems, two of the fundamental



(a) Female Speaker.

(b) Male Speaker.

Figure 11: Spectrogram of speech data for two speakers. From top original broadband speech sampled at 16 kHz, narrowband 8 kHz speech and 16 kHz bandwidth expanded speech reconstruction was performed using 133 basis with $\delta = 0.1$ and $\epsilon = 0.4$ and frame size of 25 ms.

problems with these subjective tests are, (1) how to manage and set the expectations of a listener as to what constitutes a glitch or an artifact and, (2) how to consistently generate baseline distorted speech for quality comparisons.

To overcome the problems mentioned previously, we needed a speech distortion system that would produce consistent, measurable, and reproducible distortions in clean utterances. The intended system should also provide control on the rate at which glitches are introduced in speech.

Figure 12 shows the block diagram of such a system that we designed to introduce controlled artifacts, glitches and noise in the broadband speech. This system is based on a LPC analysis-synthesis filter. Artifacts are introduced in broadband speech by jittering the LP gain ($G = G + J$). The mismatch in gain of consecutive frames manifests as glitches in synthesized audio. The jitter (J) is a zero mean, 0.5 variance normal random variable. This system also provides a *jitter control* that regulates the % of LPC frames with gain mismatch. Additive noise can also be introduced in the synthesized speech during post processing.

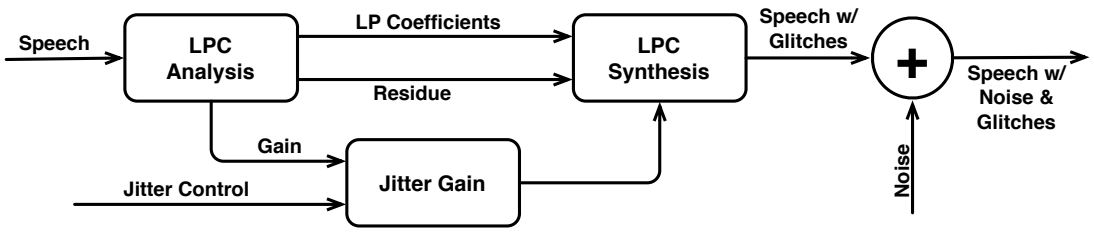


Figure 12: Block diagram of artifacts generation system.

Subjective tests were designed to answer following two questions:

1. Which audio would a subject prefer: 8 kHz sampled narrowband speech or ABE speech?

2. What is the overall quality of synthesized broadband speech? (The quality of an utterance is impacted by the presence of glitches, distortion and noise.)

Following seven comparative tests were designed to evaluate the subjective quality of synthesized broadband speech:

1. ABE vs. Narrowband speech (8K).
2. ABE vs. Broadband speech with 1% frames with gain mismatch (G 1%).
3. ABE vs. Broadband speech with 10% frames with gain mismatch (G 10%).
4. ABE vs. Broadband speech with machine gun noise at 15 dB SNR (MC GUN).
5. ABE vs. Broadband speech with white noise at 10 dB SNR and 5% frames with gain mismatch (White + G5).
6. ABE vs. Broadband speech with white and machine gun noise at 15 dB SNR (White + MC).
7. ABE vs. Broadband speech with 30% frames with gain mismatch (G 30%).

Twenty-five subjects participated in these tests. Each test session lasted no more than 30 minutes. The participants were allowed to take a break during the test to reduce listener fatigue. All the subjects used the same Sony MDR-V600 circumaural headphones. Identical test instructions were provided to all the subjects. Each test consisted of 28 pairs of utterances: four pairs (2 male and 2 female speakers) of utterances for each kind distortion.

Each test consisted of a pair of utterances. Subjects could replay utterances before making a decision. Upon completing the playback, the subject was asked a question "How is the quality of second utterance as compared to first utterance?". The answers were recorded on a five point-scale, 1 being worst and 5 a strong preference for the second utterance. A score of 3 indicated that there was no difference in quality of the two test utterances.

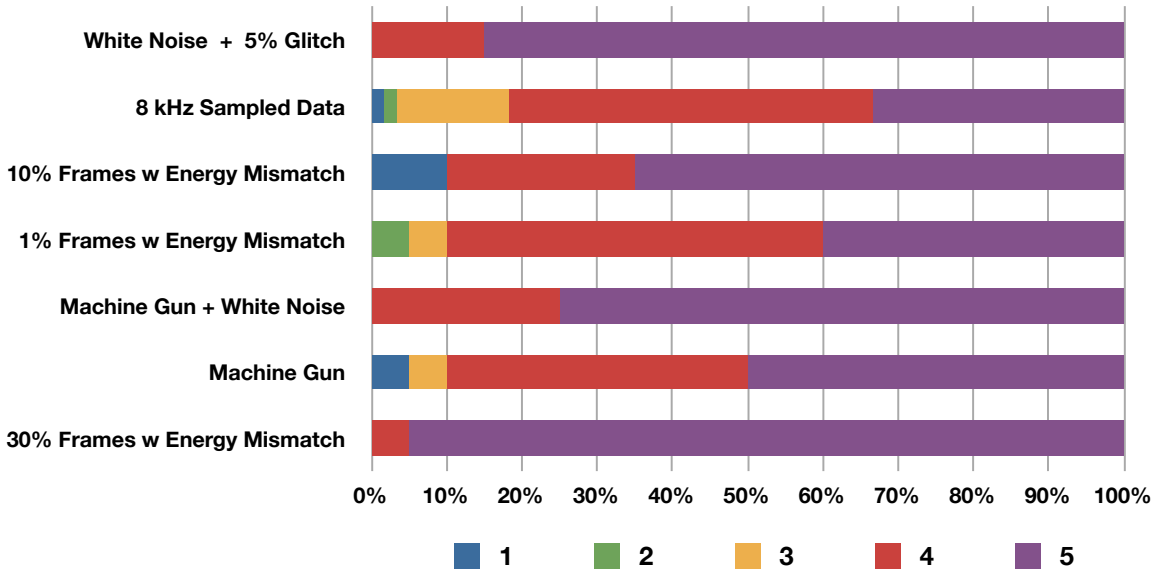


Figure 13: Subjective test preference scores indicating the users choice of ABE signal over narrowband/glitch speech.

Figure 13 shows the score distribution for the subjective tests. More than 80% of the listeners preferred ABE broadband utterance to the narrowband speech. The cases where the performance of the ABE was poorer than that of narrowband speech constituted less than 3% of the test utterances.

The second result emerges from of the comparison of ABE with 1% Glitch (G 1%) utterances. More than 90% concur that the ABE speech has less than 1% of it frames in error. The polling for other glitch cases (G 10% and G 30%) shows a complete bias of the listeners towards ABE speech. This result establishes that not a single synthesized utterance has more that 10% of frame with annoying artifacts. The 1% and 10% glitch tests provide us with metric to set lower and upper limit on rate at which artifacts are introduced in synthesized speech.

The third set of tests measure the amount of noise introduced by the process of bandwidth expansion. From the preference scores for tests cases such as MC GUN, White + MC, and White + G5, we can infer that the PSMAP-based ABE does not introduce a significant amount noise in the synthesized speech. Further in worst

cases where noise and distortion is present the SNR of synthesized speech does not fall below 15 dB.

Table 3 presents a comparison between PSMAP, vector quantizer (VQ) and Jax and Vary (J&V)[45] based ABE systems. The VQ based, speaker independent system, used a vector quantizer with a codebook of size 2048. This codebook encodes a one-to-one mapping between the narrowband and broadband spectra. The J&V ABE system is a statistical model based ABE algorithm that uses a hidden Markov model to map narrowband and broadband LPC, and uses a modulation technique to extend the narrowband LPC residue. The VQ based system is the simplest of the three systems, it does not use a sophisticated statistical model to generate mapping between the narrowband and broadband speech spectra. The SD performance of the VQ model is poorest of all three system (8.21 dB). The best (1024 Gaussians) speaker-dependent J&V system has the SD performance (6.85 dB) poorer than smallest (total of 100 Gaussians) speaker-independent PSMAP (4.19 dB) system.

Table 3: Comparison of ABE systems

Methods	**# of Gaussians**	**SD (dB)**
VQ Best (Speaker Independent)	2048	8.21
J&V Best (Speaker Independent)	1024	6.85
J&V Best (Speaker Dependent)	1024	5.90
Smallest PSMAP (Speaker Independent)	100	4.19

3.4 Conclusions

In this chapter we presented an application of the probabilistic model of speech production. PSMAP were successfully applied towards the task of ABE. The ABE system presented in this chapter out-performs the current state-of-the-art ABE system. A battery of comprehensive objective and subjective tests show that PSMAP-based ABE system successfully reconstructs missing high frequency components. The quality of synthesized speech is also good and in the worst case, the listeners agree that the

synthesized speech has less than 1% frames with artifacts and the SNR of synthesized speech never falls below 15 dB.

Measuring and quantifying artifacts is one of the fundamental problems faced by speech researchers. In this chapter we proposed an artifact generator system as a solution to this problem. The artifact generator allows user to introduce controlled glitches and background noise into undistorted audio. The degraded speech can be used to set up a known baseline for listening tests.

CHAPTER IV

CONSTRAINT PSMAP AND SPEECH ENHANCEMENT

4.1 Introduction

Enhancement of speech is one of the most important steps in the processing of speech for compression, modification, or recognition. Much of the recorded speech is corrupted by additive noise, which depends on the environment where the recordings were made. Noisy speech is difficult to perceive and it causes problems for vocoders [80]. To mitigate noise, either the recordings must be made in a studio, or significant pre-processing must be performed before speech is compressed, stored, or played. If the speech is recoded for an automatic speech recognition task (ASR), noise is a major enemy [62, 74, 34, 51]. Devices and applications that gather audio for ASR is rapidly increasing as cloud-based ASR systems are emerging. The largest groups of consumer devices that have an ASR-based application are mobile phones. Recordings made on such devices are plagued by variety of noises dependent on the environment. Considering the plethora of ASR speech applications taking root in consumer market, good speech enhancement algorithms have become vital.

Given the diversity of the applications and environments, enhancement algorithms must work on speech degraded by many different noises. In this chapter we will assume that the noise is additive and statistically independent of the speech, which is a relaxation from reality. In addition we also assume that the noise data/statistics are not independently available. That is to say, the recordings are made using a single microphone and the noise and speech statistics are to be estimated from the same data. Increasingly, multichannel systems are gaining popularity in consumer market. Systems such as iPhone^{TM}, Kinect^{TM}, laptops etc., do possess multiple

microphones but due to lack of dedicated software, we do not always have access to the multichannel data. This limits the application of multichannel enhancement algorithms in the consumer devices. Single channel speech enhancement is therefore an important problem and has been an area of active research for over four decades.

One of the problems that plague the enhancement community is the lack of metrics to quantify the performance of an enhancement technique. The perception of a speech signal is usually measured in terms of quality and intelligibility. The quality of speech can be quantified using a subjective or an objective measure. MOS scores for example, can be used to assigning a subjective quality metric to enhanced speech. One problem with this scoring system is the requirement of a large group of listeners and a standardization of the listening environment. Objective measures such as spectral distortion (SD), and segmental SNR reduction/improvement (segSNR) can be used to estimate the improvement in quality of enhanced speech. These measures, however, fail to quantify distortions and artifacts introduced by the enhancement system. Intelligibility measures the percentage of words correctly identified by a listener in a controlled fashion. Intelligibility score, however, completely overlooks the presence of artifacts, which impact the quality without affecting the intelligibility.

These two metrics of speech quality measurement are not correlated and we can improve the intelligibility at the cost of quality. It is seldom that we have algorithms that improve both perceptual quality and ASR accuracy. Front-end algorithms that enhance speech usually do not provide equivalent improvement in intelligibility or accuracy of the ASR system. From an information theory point of view this phenomenon can be explained by the data processing theorem [16]. Consider that s is the source, w is the noisy data, d is a process/algorithm and r is the enhanced data; then according to the data processing theorem "the information that r conveys about the source s is less than or equal to the information conveyed by w". Simply speaking this theorem states that a degraded signal will provide us more information about

the clean signal than that provided by enhanced signal. Consequently, ASR system will have better word accuracy if they use noisy speech.

Since noise is a naturally occurring random process, speech enhancement is a problem of statistical estimation of a signal from the sum of two random processes. To solve this estimation problem, theory dictates that we need models for the signals and a distortion measure, which in turn needs to be optimize.

Choice of a robust statistical model is the first problem faced by speech researchers. Over the years, various statistical models have been proposed for the speech signal [11, 22, 23, 10], each with its benefits and drawbacks. For e.g., Chen and colleagues [10] suggested that the STFT of speech can be modeled with a Laplace distribution, but algorithms with this statistical model do not have a closed-form solution unless approximations are made. It can be demonstrated heuristically that the magnitude spectrum of speech is not normally distributed, in spite of this fact studies such as [22] have used the central limit theorem is used to justify the choice of a Gaussian distribution for the STFT spectra of speech.

The quasi-stationary nature of speech also complicates the enhancement filter design. Any algorithm proposed to enhance speech should be fast enough to track changes in the spectrum of speech as well as noise without sacrificing quality.

Choice of a distortion measure is extremely important in designing a speech enhancement algorithm. The third problem that plagues speech enhancement research is the lack of perceptually relevant distortion measure.

Rightfully so, the focus of speech enhancement research has been on either proposing statistical models for speech spectra, or proposing novel and perceptually-relevant distortion measures.

4.2 Single Channel Speech Enhancement

Let $x(t)$ denote the clean speech signal that we are interested in estimating. Unless we are recording in a sanitary environment such as in a soundproof anechoic studio, the microphone picks up unwanted noise $n(t)$ and records it along with the speech. Let $y(t)$ denote this noisy signal (Equation (28)):

$$y(t) = x(t) + n(t). \tag{28}$$

Equation (28) provides a simple view of the acquired signal, in reality microphone nonlinearities, echoes, room reverberations, etc. mar the final acquired utterance. Equation (29) indicates the speech acquired in presence of nonlinearities and noise.

$$y(t) = h(t) * x(t) + n(t), \tag{29}$$

where $h(t)$ is response of the channel, which is the combined response of all nonlinear sources (e.g., microphone, echoes, room reverberations) in the system.

Channel and the other nonlinearities cause havoc on an ASR system [6, Ch. 33]. Simple and complex algorithms exist to mitigate the channel effects. This thesis will not deal with compensating nonlinearities, but we will propose methods to counter additive noise $n(t)$.

The human auditory system is relatively insensitive to noise in the phase spectrum [84]. This provides a great incentive to design and implement the speech enhancement systems in the frequency domain. As the perceptual improvement is only related to the magnitude spectrum, a noise suppressor must just shape/weight the magnitude spectra to enhance speech. The noise suppression filters described throughout this chapter will be designed in the frequency domain to avoid numerical instabilities, and to control non-negativity of the magnitude spectrum.

Let $X(k,l)$, $N(k,l)$ and $Y(k,l)$ be the l^{th} discrete Fourier transform (DFT) coefficients for the k^{th} frame of clean speech, noise, and noisy speech respectively. Using

this notation, the speech enhancement problem can be succinctly stated as that of finding an estimator that minimizes the conditional expectation of distortion $\mathcal{D}$ given the noisy signal (the bin and frame indices l and k have been dropped for brevity).

$$\hat{X} = \underset{\hat{X}}{\operatorname{argmin}}\ \mathbf{E}\{\mathcal{D}(X, \hat{X})|Y\}. \tag{30}$$

A general solution to the speech enhancement problem is in Equation (31):

$$\widehat{X} = G(\xi, \eta)Y, \tag{31}$$

where G is the gain of the denoising filter, computed by optimizing the distortion measure. The filter gain, G, is a function of $\xi = \left(\frac{X}{N}\right)^2$ defined as the *a priori* SNR and $\eta = \left(\frac{Y}{N}\right)^2$ defined as the *a posteriori* SNR. This convention of defining *a priori* and *a posteriori* SNR is adapted from McAulay & Malpass [61].

Table 4 lists a variety of Gain functions proposed over the years. The form of the suppression gain (G) depends on the choice of statistical model for the speech spectra, and the distortion function. As an example, the suppression gains for the two Ephraim and Malah estimators listed in the Table 4 are different even though they both assume a Gaussian model for distribution of the STFT coefficients. The reason for the discrepancy in gain is the choice of the distortion measure, one optimizes the MSE and the other optimizes the log-MSE.

4.2.1 Model Selection

The statistical model used to represent the distribution of magnitude spectrum of speech (X) plays an important role not only in determining the quality of enhanced speech but also impacts the analytical solution of Equation (30). Generally speaking, the models are selected for computational tractability. The Gaussian distribution is widely promoted and used as the statistical model for magnitude spectra of speech [22, 22, 61] as the algorithms derived using the Gaussian assumption often lead to tractable solutions. The Gaussian assumption has been challenged by many authors [60, 67, 10]

Table 4: Gain function for conventional speech suppression rules.

Method	Gain(G)
Spectral Subtraction [8]	$\left(\frac{\xi}{\xi+1}\right)^{\frac{1}{2}}$
MMSE (Wiener)	$\frac{\xi}{\xi+1}$
ML (McAulay & Malpass) [61]	$\frac{1}{2}\left[1+\left(\frac{\xi}{\xi+1}\right)^{\frac{1}{2}}\right]$
STFT MMSE(Ephraim & Malah) [22]	$\frac{\sqrt{\pi\nu}}{2\eta}\left[(1+\nu)I_0\left(\frac{\nu}{2}\right)+\nu I_1\left(\frac{\nu}{2}\right)\right]\exp\left(\frac{\nu}{2}\right)$
log-STFT MMSE(Ephraim & Malah) [23]	$\left(\frac{\xi}{\xi+1}\right)exp\left(\frac{1}{2}\int_{\nu}^{\infty}\frac{e^{-t}}{t}dt\right)$
JMAP SAP [85]	$\frac{1}{2(1+\xi)}\left[\xi+\sqrt{\xi^2+2(1+\xi)\frac{\xi}{\eta}}\right]$
MAP SAE [85]	$\frac{1}{2(1+\xi)}\left[\xi+\sqrt{\xi^2+(1+\xi)\frac{\xi}{\eta}}\right]$
MMSE SP [85]	$\sqrt{\frac{\xi}{\xi+1}\left(\frac{1+\nu}{\lambda}\right)}$

over the years, some authors have suggested Gamma or Laplace distributions [67, 59, 11] as alternative. Often these model assumptions produce intractable solutions unless some allowances are made to simplify the problem. These relaxations often do not preserve the Laplacian/Gamma distributions and thereby make the initial choice moot.

4.2.2 Distortion Measures

The second focus of noise suppression research has been the selection of a distortion measure $\mathcal{D}$. The most popular distortion measure is the mean square error (MSE). When combined with the Gaussian assumption, it produces the very popular Wiener filter. It has been argued that MSE does not have a strong perceptual significance. This led Ephraim and colleagues [23] to use a log-MSE as the distortion measure for the spectrum, which is perceptually more meaningful. The underlying assumption for using a log compression function is to better approximate the human auditory response.

A better way of mimicking the auditory response is to design the filter using a mel-warped spectrum [36, 26]. Instead of estimating the gain for every DFT bin,

a mel-warped Wiener filter estimates the gain for only the mel-spaced bands. The filters using this distortion measure tend to perform better than those using simple log-distortion counterparts, as we will see later in this chapter.

4.2.3 Typical Speech Enhancement System

Figure 14 shows the block diagram of a typical speech enhancement system. As the noise suppression is only performed on the magnitude spectra; phases of the noisy speech are directly used to synthesize cleaned speech.

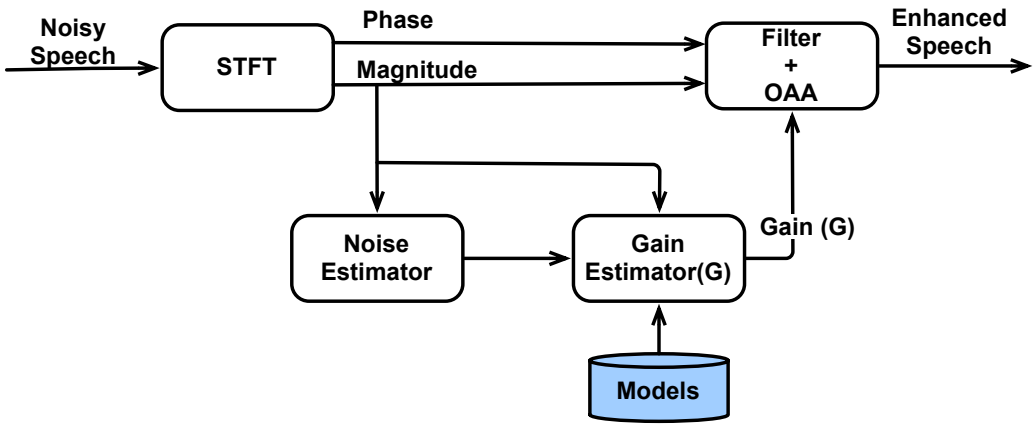


Figure 14: Block diagram of speech enhancement system

A traditional speech enhancement system consists of three components:

1. Front and back ends: include the STFT blocks, the filter implementation, and the overlap and add (OAA) reconstruction blocks. The STFT block is responsible for buffering and windowing of the incoming signal. The STFT block also estimates the magnitude and phase spectrums of each frame.

2. Noise estimator: This is the most important block of the system. The noise estimator consists of a voice activity detector (VAD). The labels generated by VAD are used for estimation/update of the noise statistics. An accurate

estimator of the noise spectrum is the backbone of a robust noise suppressor. This block is often closely integrated with the gain estimator.

3. Gain estimator: This block uses the estimate of the power spectrum of noise and speech provided by the front-end and noise estimator to calculate η and ξ. The estimate of *a posteriori* and *a priori* SNRs are then used to compute the noise suppression gain G. The gain can be either directly applied to the magnitude spectrum of noisy speech, or used to estimate the taps of an enhancement filter. The algorithm used in this block depends on both the choice of statistical model of the magnitude spectrum and on the choice of the distortion measure to be minimized.

All the suppression rules listed in Table 4 need an estimate of *a priori* SNR to compute the gain. To compute the *a priori* SNR the system must have access to the power spectra of "clean speech". The estimation of clean speech is the goal of a noise suppression system.

Estimating *a priori* SNR without access to the spectrum of clean speech is a conundrum faced by all speech enhancement systems. The success of any noise suppressor is directly tied to its ability of estimating *a priori* SNR. In this chapter we will present a PSMAP-based *a priori* SNR estimator and compare its performance to existing *a priori* SNR estimators.

4.3 The Chicken or the Egg?

To understand the importance of *a priori* SNR, we must first study its impact on the performance of a noise suppression system. To this effect we set up a noise suppression system similar to one described in Section 4.2.3. This system was designed to use Wiener gain to suppress noise. The statistics of the noise were estimated using a threshold based VAD (Appendix C). The baseline performance of the noise suppression system was computed using a Wiener filter that had access to the true

spectrum of clean speech. We call this particular filter as the oracle Wiener filter (OWF). The OWF was able to access the true *a priori* SNR (ξ_o), and hence has the best performance. The system performance is measured using two objective metrics:

- **Spectral Distortion**: A parameter that measures the deviation log spectrum of enhanced speech to the log spectrum of the original clean speech. Equation (32) illustrates the method used for computing SD:

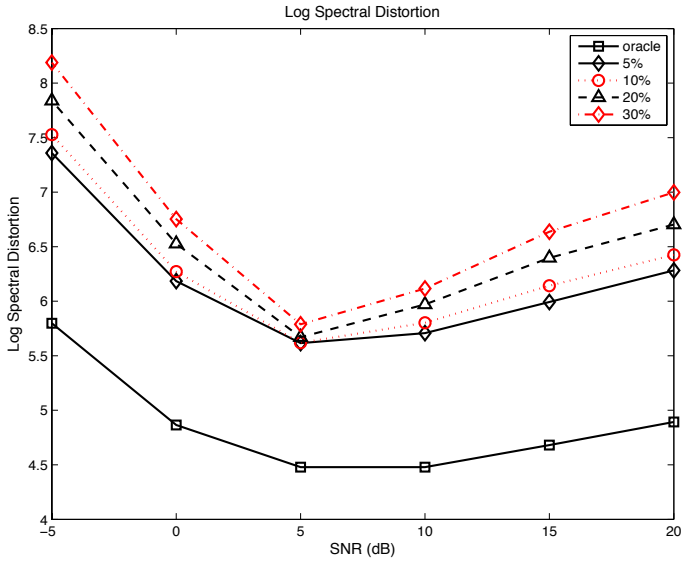
$$SD = \frac{20}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} \left| \left(\log_{10} |X(k,l)| - \log_{10} |\widehat{X}(k,l)| \right) \right|. \tag{32}$$

- **Improvement in Segmental SNR value (segSNR)**: This parameter measures the reduction in segmental SNR achieved by the suppression rule. The improvement in segSNR is obtained by computing the difference in segmental SNR values of the noisy speech and the enhanced speech. The equations (33) – (35) show the procedure to compute segSNR:

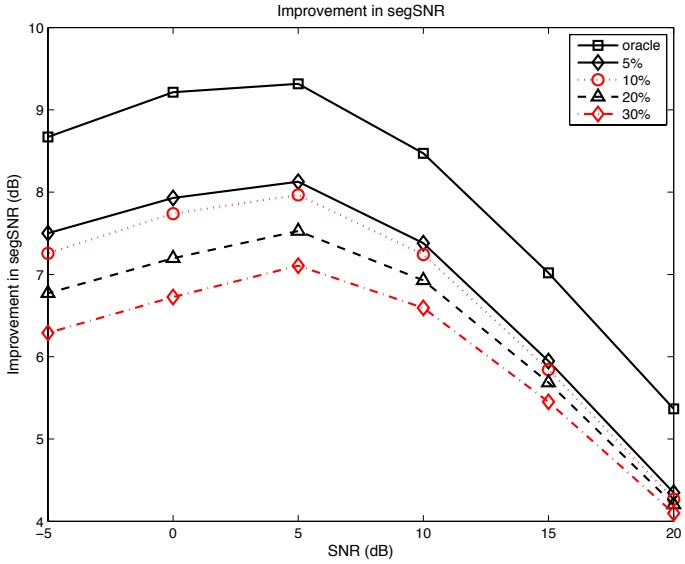
$$\begin{aligned} SNR_{cleaned} &= 10 \log_{10} \Big(\frac{\sum_{i=1}^{F} \hat{x}(i)^2}{\sum_{i=1}^{F} n(i)^2} \Big), & (33) \\ SNR_{noisy} &= 10 \log_{10} \Big(\frac{\sum_{i=1}^{F} y(i)^2}{\sum_{i=1}^{F} n(i)^2} \Big), & (34) \\ segSNR &= SNR_{cleaned} - SNR_{noisy}. & (35) \end{aligned}$$

We then perturbed the true *a priori* SNR was by a small amount ($\pm\delta$). The incorrect values of *a priori* SNR ($\xi_d = \xi_o \pm \delta$) were used to estimate new Wiener filters. The performance of the new WF was compared to the OWF. SD and segSNR were used as metrics for comparison.

Figures 15(a) and 15(b) document the impact of inaccurate estimate of *a priori* SNR on the performance of Wiener filter-based suppression system. Incorrect estimation of a priori SNR severely degrades the objective performance of the noise



(a) Log Spectral Distortion



(b) Reduction in segmental SNR

Figure 15: The impact of incorrect *a priori* SNR estimation on the performance metrics. The legend indicates the % displacement of *a priori* SNR from the true value.

suppression system. Even a small 5% error in estimation of *a priori* SNR degrades SD and segSNR scores by over 2 dB (Figures 15). The impact of incorrect *a priori* SNR is more severe at low SNR conditions than at high SNR conditions. Therefore, the *a priori* SNR estimator is the most critical block in the complete noise suppression

system.

The *a priori* SNR depends on the estimates of speech and noise spectra. The estimate of *a priori* SNR can be affected either by the incorrect estimation of noise spectrum, or by the incorrect estimation of the clean speech spectrum. Given a state-of-the-art noise estimator, an accurate estimator speech spectrum becomes the most crucial part of the noise suppressor.

4.3.1 Estimating the *a priori* SNR

Spectral subtraction is one of the earliest and the simplest methods of *a priori* SNR estimation. Spectral subtraction schemes use the estimate of the noise spectrum to generate the estimate of clean speech spectrum, as shown by equations (36) - (38)

$$X = \max(Y - N, \widehat{X}_{min}), \tag{36}$$

$$\frac{X}{N} = \max\Big(\frac{Y}{N} - 1, \frac{\widehat{X}_{min}}{N}\Big), \tag{37}$$

$$\xi = \max(\eta - 1, \xi_{min}). \tag{38}$$

The parameter ξ_{min} is the floor of the *a priori* SNR (typical value of the ξ_{min} is between -20 dB to -50 dB). Flooring the SNR value guarantees the stability of the suppression filter by forcing the power spectra to be non-negative. Flooring the minimum *a priori* SNR to a non-zero value also helps reduce musical or tonal noise in the reconstructed speech. There are multiple problems with such an estimator. One of the bigger problems is that the accuracy of the estimated speech spectrum is directly tied to accuracy of the noise estimator. Specially, at low SNR condition the estimate of clean speech spectrum is so poor, that most of the *a priori* SNR estimates are usually floored to ξ_{min}. This repeated flooring of *a priori* SNR causes the filter to degenerate.

Ephraim and Malah [22], proposed a decision-directed (DD) *a priori* SNR estimator. The DD *a priori* SNR estimator has proven to be extremely useful when combined with a good suppression rule. The DD *a priori* SNR (Equation (39)) is

computed using a weighted sum of two components: one part of the estimate is based on the *a posteriori* SNR of the current frame (η_n), and the second is based on suppressor gain ($\mathbf{G}_{n-1}$) and the *a posteriori* estimate (η_{n-1}) from the previous frame.

$$\widehat{\xi_n} = \alpha \mathbf{G}_{n-1}^2 \eta_{n-1} + (1-\alpha) \cdot \max(\eta_n - 1, \xi_{min}). \tag{39}$$

The parameter α ($0 \leq \alpha \leq 1$) is the weighting parameter. The choice of α will have a significant effect on the performance of the suppression system. Unfortunately, the DD estimator is a very nonlinear function, which makes it very difficult to optimize for α analytically. The value of the weight parameter is often heuristically determined, even for the simplest of gain functions such as the Wiener filter. In our experiments, we found that the values of α in the range of 0.95 to 0.98 gave the best performance for the objectives under consideration. The DD *a priori* SNR estimator is computationally efficient, and it significantly improves the performance of a suppression system independent of the gain function being used. The exponential window (controlled by α) used in the DD estimator dampens random fluctuation in the estimate of *a priori* SNR; this dampening in-turn reduces the artifacts in the cleaned audio.

The DD *a priori* SNR estimator described in Equation (39) with some modifications can be rewritten as Equation (40)

$$\widehat{\xi_n} = \alpha \widehat{\xi_{n-1}} + (1-\alpha) \cdot \max(\eta_n - 1, \xi_{min}). \tag{40}$$

Best performance of the DD *a priori* SNR estimator is often observed for values of α close to unity. In such a scenario the estimate of *a priori* SNR (ξ_n) for the current frame is mostly dominated by the *a priori* SNR (ξ_{n-1}) of the previous frame. So in cases where the *a priori* SNR varies rapidly, the DD estimator will have problems catching up to the true SNR.

The DD estimator rely more on the past measurements. It ignores the noise conditions of current frame as indicated by the measurements of *a posteriori* SNR

(η_t). The *a posteriori* SNR for the current frame ($\max(\eta_n - 1, \xi_{min})$) serves as a correction parameter, which only comes into play if *a priori* SNR is extremely low.

The DD *a priori* SNR was a major breakthrough in the field of speech enhancement. Further the simple implementation of the system led to its widespread acceptance in almost all systems. The DD estimator does have its fair share of problems, much of the problems are associated with the weight parameter (α). The DD *a priori* SNR estimator has trouble adapting to fast-varying noisy conditions due to use of the damping parameter α. Using small values of α is one way of speeding up the tracking of rapidly varying *a priori* SNR. Using small values of α encourage the DD estimator to base the *a priori* SNR estimate on the *a posteriori* SNR of current frame. The performance of DD estimator (with small values of α) is excellent at high SNR conditions but at low SNR conditions the estimator breaks down and most of the *a priori* SNR estimates are the floor values (ξ_{min}). To decrease the transient nature of the DD estimator, Hassan and colleagues in [38] suggested an algorithm that adapts the values of α conditioned on the current estimate of SNR.

To combat the delay in estimation of the *a priori* SNR, Cohen [14, 13] proposed a modification to the existing DD *a priori* SNR estimator. This new estimator is a non-causal estimator that looks ahead a couple of frames to compute an intermediate *a priori* SNR estimate. This intermediate estimate is propagated through the standard DD estimator. As we will see later in the chapter, this does marginally improve the objective performance of the system.

Another group of methods uses data driven approach to estimate *a priori* SNR. Fingscheidt [28, 27] proposed several data driven techniques to estimate the weighting (α). The weight depends on the current values of *a priori* and *a posteriori* SNR. The rules for calculating α are precomputed from the training data. Erkelens [24] used similar methods to propose weighting rules, in his case the models were trained

under white noise conditions with known variance. Suhadi [79] used a neural network to build a model that established a relationship between *a posteriori* SNR and *a priori* SNR from the training speech data. Most of the data driven *a priori* SNR estimators require synchronized clean and noisy (stereo data) utterances for training the models. Further these methods tend to perform poorly in conditions where operating noise does not match training data.

The previously suggested data-driven schemes [28, 27, 24] cannot be used in the absence of stereo training data. Also the existing data-driven schemes tend to perform poorly in conditions where the operating noise conditions do not match the noise in training dataset.

We therefore, need an *a priori* SNR estimator that is fast enough to track and respond to rapid changes in *a priori* SNR, and an estimator whose performance is not dependent on tuning of a single parameter (α). In this chapter we will present a method that leverages the PSMAP-based speech production model to estimate the *a priori* SNR from noisy speech measurements. The PSMAP-based *a priori* SNR estimator also falls under the umbrella of data driven *a priori* SNR estimators, but unlike the other schemes we only need access to clean speech during the training phase.

The PSMAP-based scheme does not incorporate any noise information in the model. The PSMAPs are always adapted to the noise in the operating environment. This property of the PSMAP-based estimator is attractive because the performance of the estimator is not tied to the noise in training data.

Before presenting the *a priori* SNR estimator we will take a slight detour to present some enhancement to the PSMAPs proposed in the previous chapter.

4.4 Constraint Probabilistic Space Maps

The PSMAP presented in Chapter 2 exploited the many-to-one mapping inherently present between the VT and speech spectrum. The sparse version PSMAP further imposes sparsity constraints on the mapping between the two subspaces using entropic priors. The sparse/simple PSMAPs however, do not impose any constraints on the state transition within the individual subspaces, as a result simple PSMAPs do not exploit the temporal nature of speech. In reality, consecutive transitions between the latent states of a subspace modeling the VT are neither uniform nor unconstrained.

A human subject can speak at a finite speed and has a vocal tract with finite flexibility. These anatomical constraints impose physical limits on the latent states of VTAF subspace. The implications of the anatomical constraints on the states of a PSMAP are listed below:

- **Constraint 1:** The areas of adjacent segments in the uniform lossless tube model cannot vary by an arbitrary large amount (flexibility restriction).
- **Constraint 2:** The VT area profile across consecutive frames cannot vary arbitrarily rapidly.

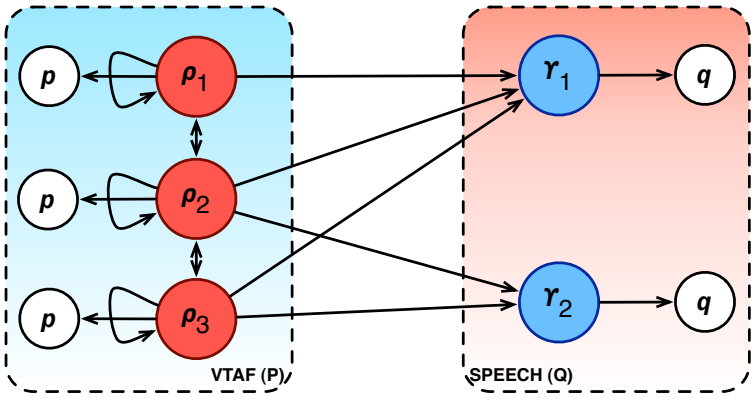


Figure 16: Graphical model for a constraint probabilistic space map.

The anatomical constraints manifest as a restriction on temporal transitions between the states ρ_m of the Subspace $\mathcal{P}$. To accommodate the temporal constraints,

Subspace $\mathcal{P}$ will be modeled with hidden Markov model (HMM). Figure 16 shows the graphical model for constraint-PSMAP (CPSMAP).

4.4.1 The Model

Figure 16 shows the graphical model for an improved PSMAP. This model improves upon the simple PSMAP by imposing temporal constraints on the VTAF subspace.

The symbols ρ and γ represent the hidden states that model Subspace $\mathcal{P}$ and Subspace $\mathcal{Q}$ respectively. Subspace $\mathcal{P}$ and Subspace $\mathcal{Q}$ are modeled with N and M distinct latent states. The states of the Subspace $\mathcal{P}$ form a first order hidden Markov chain (HMM) where the transition from a current state ρ_t to ρ_{t+1} (t is the time instant) is governed by set of probabilities $\mathbf{A}^{\rho}$ such that $p(\rho_t = j|\rho_{t+1} = i) = a_{ij}^{\rho}$, where $\sum_j a_{ij}^{\rho} = 1$. Each of the states of Subspace $\mathcal{P}$ is modeled with a Gaussian mixture, with L Gaussian components per mixture as shown in Equation (41):

$$p(\mathbf{p}_t|\rho_n) = \sum_{l=1}^{L} w^{n,l}\mathcal{N}(\boldsymbol{\mu}_{\rho}^{n,l}, \boldsymbol{\sigma}_{\rho}^{n,l}), \tag{41}$$

where $w^{n,l}$, $\boldsymbol{\mu}_{\rho}^{n,l}$ and $\boldsymbol{\sigma}_{\rho}^{n,l}$ are the weights, means and variances of the l^{th} Gaussian of the n^{th} state.

The Subspace $\mathcal{Q}$ is mapped with M Gaussians $\mathcal{N}(\boldsymbol{\mu}_{\gamma}^{m}, \boldsymbol{\sigma}_{\gamma}^{m})$ where $m = 1, 2, \ldots, M$. The Gaussians mapping both subspaces have diagonal covariances. The transition between states of Subspace $\mathcal{P}$ and Subspace $\mathcal{Q}$ are encoded in a transition matrix $\mathbf{A}$, where $a_{mn} = p(\gamma_m|\rho_n)$ and $\sum_m a_{mn} = 1$.

4.4.2 Parameter Estimation using EM

Each subspace of the model has a different structure, and different set of parameters. The Subspace $\mathcal{P}$ is modeled with a HMM which has three parameters to estimate, $\Lambda_{\rho} = \{\pi, \mathbf{A}^{\rho}, p(\mathbf{p}_t|\rho_m)\}$, where π are the initial probabilities distribution over the states of Subspace $\mathcal{P}$. Subspace $\mathcal{Q}$ is modeled with M Gaussians and has two parameters per Gaussian to estimate. Estimation of these parameters is performed using standard expectation-maximization (EM) [19, 7] algorithm. The standard EM

algorithm has following two steps:

1. E-step: Compute the *a posteriori* probabilities:

$$p(\mathbf{p}_t, \boldsymbol{q}_t|\rho_n, \gamma_m) = \frac{p(\mathbf{p}_t, \boldsymbol{q}_t, \rho_n, \gamma_m)}{\sum_{m=1}^{M}\sum_{n=1}^{N} p(\mathbf{p}_t, \boldsymbol{q}_t, \rho_n, \gamma_m)}, \tag{42}$$

   where the joint probability over the observed and latent states is given by Equation (43):

$$p(\mathbf{q}_t, \mathbf{p}_t, \gamma_m, \rho_n) = p(\mathbf{q}_t|\gamma_m)p(\gamma_m|\rho_m)p(\mathbf{p}_{(1:t)}, \rho_{(1:t)}). \tag{43}$$

   As Subspace $\mathcal{P}$ is modeled with a HMM, the term $p(\mathbf{p}_{(1:t)}, \rho_{(1:t)})$ is the probability of observing a sequence of $\mathbf{p}$ upto time instant t. This probability is computed using the forward-backward algorithm for Markov chains [7, 69].

2. M-step: Maximize the complete data likelihood $\mathcal{L}$ to estimate parameters of the model:

$$\mathcal{L} = \mathbf{E}_{\Lambda_\rho, \theta|\mathbf{q},\mathbf{p},\Omega}\{\log p(\mathbf{q}_t, \mathbf{p}_t, \rho_n, \gamma_m)\}, \tag{44}$$

   where $\Omega = \{\Lambda_\rho, \mathbf{A}, \gamma\}$, are the complete model parameters.

Parameter estimation is performed by alternatively solving Equations (42) and (44).

Solving for the parameters of the Gaussians in Subspace $\mathcal{Q}$ yields:

$$\boldsymbol{\mu}_\gamma^m = \frac{\sum_{t=1}^{T}\sum_{n=1}^{N} p(\gamma_m, \rho_n|\mathbf{q}_t, \mathbf{p}_t)\mathbf{q}_t}{\sum_{t=1}^{T}\sum_{n=1}^{N} p(\gamma_m, \rho_n|\mathbf{q}_t, \mathbf{p}_t)} \tag{45}$$

$$\boldsymbol{\sigma}_\gamma^m = \frac{\sum_{t=1}^{T}\sum_{n=1}^{N} p(\gamma_m, \rho_n|\mathbf{q}_t, \mathbf{p}_t)(\mathbf{q}_t - \boldsymbol{\mu}_\gamma^m)^2}{\sum_{t=1}^{T}\sum_{n=1}^{N} p(\gamma_m, \rho_n|\mathbf{q}_t, \mathbf{p}_t)} \tag{46}$$

The parameters of the Subspace $\mathcal{P}$ (HMM) are computed using a modified version of Baum-Welch method [7]. The difference between the traditional Baum-Welch and the one used here is in the number of latent states in the sufficient statistics. In the traditional Baum-Welch the sufficient statistics is the function of a single latent state, whereas, for a CPSMAP the sufficient statistic depends on a pair of latent state.

The M-step for estimation of the parameters of the Subspace $\mathcal{P}$ is given by set of equations (47) through (51).

$$\pi_n = \frac{\sum_{m=1}^{M} p(\gamma_m, \rho_n|\mathbf{q}_t, \mathbf{p}_t)}{\sum_{i=1}^{M}\sum_{n=1}^{N} p(\gamma_m, \rho_n|\mathbf{q}_t, \mathbf{p}_t)} \tag{47}$$

$$a_{ns}^{\rho} = \frac{\sum_{t=1}^{T-1}\sum_{m=1}^{M} p(\rho_n, \rho_s, \gamma_m|\mathbf{q}_t, \mathbf{p}_t)}{\sum_{t=1}^{T-1}\sum_{m=1}^{M}\sum_{n=1}^{N} p(\rho_n, \gamma_m|\mathbf{q}_t, \mathbf{p}_t)} \tag{48}$$

$$w^{n,l} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n^l, \gamma_m|\mathbf{q}_t, \mathbf{p}_t)}{\sum_{t=1}^{T}\sum_{l=1,m=1}^{L,M} p(\rho_n^l, \gamma_m|\mathbf{q}_t, \mathbf{p}_t)} \tag{49}$$

$$\boldsymbol{\mu}_{\rho}^{n,l} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n^l, \rho_m|\mathbf{q}_t, \mathbf{p}_t)\mathbf{p}_t}{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n, \gamma_m|\mathbf{q}_t, \mathbf{p}_t)} \tag{50}$$

$$\boldsymbol{\sigma}_{\rho}^{n,l} = \frac{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n^l, \gamma_m|\mathbf{q}_t, \mathbf{p}_t)(\mathbf{p}_t - \boldsymbol{\mu}_{\rho}^{n,l})^2}{\sum_{t=1}^{T}\sum_{m=1}^{M} p(\rho_n, \gamma_m|\mathbf{q}_t, \mathbf{p}_t)} \tag{51}$$

4.5 Speech Production and Constraint Probabilistic Space Maps

The source filter model breaks speech production into two blocks: the glottal source, which provides the excitation and the vocal tract, which shapes the excitation to produce speech. It is impossible to know the real excitation and VT characteristics even while measuring the pressure or velocity at the lips of the talker. Without actual measurements, it is possible to make multiple indirect observations of both the vocal tract and excitation. The LSPs are one such observation.

The CPSMAP follow the same principles as its cousin, the simple PSMAP. The Subspace $\mathcal{P}$ models all possible configuration of the VT using N latent states. In CPSMAPs however there is a temporal constraint when we move from one state ρ_n to another state ρ_k. This temporal constraint is imposed using a HMM.

In the previous chapter, we have used LPC-MFCCs as the observations of the VT. We can still use LPC-MFCCs for speech enhancement. In case of noise suppression, PSMAPs will be used to estimate spectra of clean speech given noisy measurements. The LPC-MFCC can be easily inverted to extract the LPC spectrum of speech. The inversion of LPC-MFCC is low-rank operation, and causes unwanted smoothing of the LPC-spectrum. During our experiments we observed that the noise suppression filter produced with such a smoothened spectrum had objective performance comparable to traditional DD estimators.

Through experiments, we observed that Line spectral pairs (LSP) of a frame of speech are the better choice for observations of Subspace $\mathcal{P}$. Therefore, the CPSMAPs used for speech enhancement use LSP augmented LP gain as the observations of VT.

The Subspace $\mathcal{Q}$ is a space that models all possible spectra of speech. This subspace is modeled with M latent states. For speech enhancement we developed a special block feature set as observations of the Subspace $\mathcal{Q}$. These features were designed specifically for speech enhancement problem. The next section we will present

the specifics of block feature extraction and benefit of the new feature set over traditional spectral features.

4.5.1 Block Features for Subspace $\mathcal{Q}$

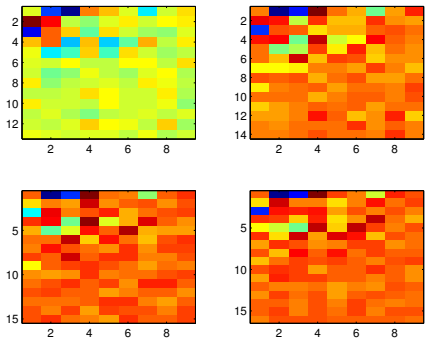
The block features proposed in this section are 2D-features that extracted from the spectrogram of the speech utterance. The feature extraction process is divided into two steps: patch extraction and decorrelation.

4.5.1.1 Patch Extraction

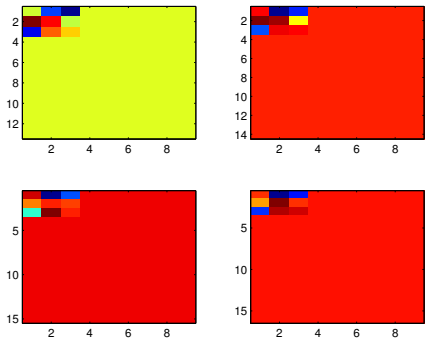
The first step in patch extraction is to compute the spectrogram of an utterance. Then at every grid point (i,k) we extract a patch of width dt and height df. In the system that we designed the grid location k is the frame number and i is the center frequency of the mel-filter bank. The width of each patch is 9 frames (i.e. 4 frames on either side of the frame being processed). This length of 9 frames corresponds to the number of frames used in computing MFCC features that are augmented with the regression velocity and regression acceleration coefficients [89]. The height of each patch corresponds to the width of each mel-filter bank at that grid location. Each patch is windowed using a 2D-Hamming window $W(df, dt)$ before further processing.

4.5.1.2 Decorrelating

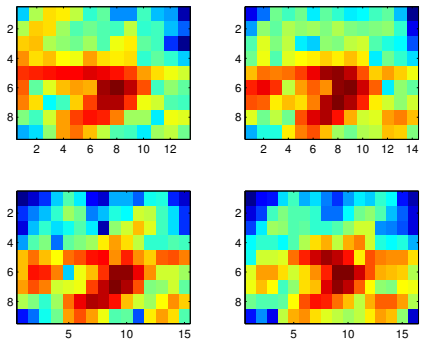
Storing and processing all the coefficients of each patch will be expensive. To reduce the computation we further process the patch to extract the final set of features for each frame. The next step is to compute 2D discrete cosine transform (DCT) of each windowed patch to produce a set of coefficients D. The 2D-DCT projects each patch onto a set of orthogonal, separable cosine basis functions that respond to horizontal speech phenomena such as harmonics and formants, vertical speech phenomena such as plosive edges, and more complex spectro-temporal noise patterns. The final step in the process is retaining only the low-order 9 coefficients. We noticed through



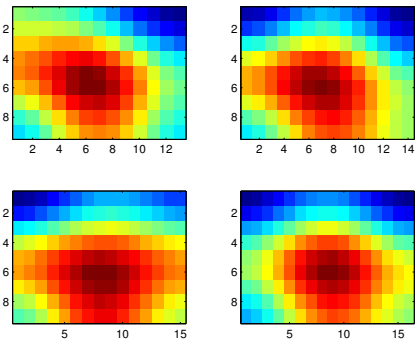
(a) DCT of 4 patchs



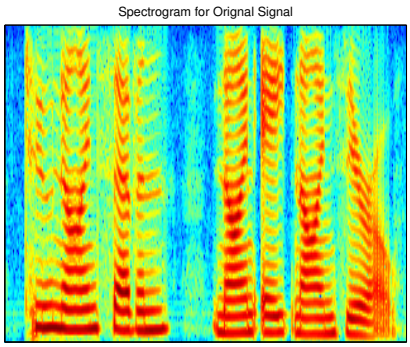
(b) 9-retained DCT coefficients



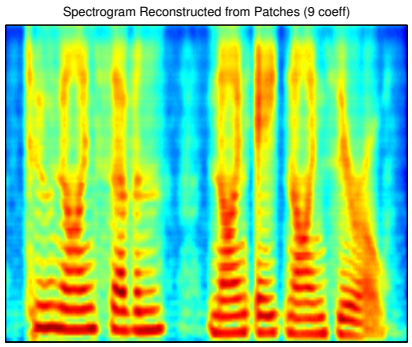
(c) Original spectrogram patch



(d) Reconstructed spectrogram patch



(e) Original spectrogram



(f) Reconstructed spectrogram

Figure 17: Patch feature extraction process through figures. Blocks in Figure 17(a) are the DCT coefficients for 4 patches of the spectrogram, the Figure 17(b) are the retained 9 low-order DCT coefficients. Blocks in Figures 17(c) and 17(d) are the reconstructed spectral patches from complete and partial DCT respectively. Figures 17(e) and 17(f) are the spectrogram reconstructed from complete and partial DCT respectively.

experiments that retaining only a few lower-order DCT coefficients does not impact the objective or subjective performance of the noise suppressor.

The effect of retaining only 9 coefficients per patch can be seen in the Figures 17(e) and 17(f). Keeping the low-order coefficients results in the smoothing of the reconstructed spectrum. The spectral smoothing results into loss of high-frequency unvoiced speech information.

One of the main reasons for using this specialized feature to represent Subspace $\mathcal{Q}$ is to isolate effect of noise to a local block. Usually at moderate SNR and high SNR conditions the lower frequency portion of the speech spectrum might not be degraded at all and possesses more information than the higher frequency region of the spectrum. Making blocks ensures degradation is isolated to a region of the subspace.

4.6 Application: CPSMAP and a priori *SNR Estimation*

To solve the problem of estimating *a priori* SNR we first build a CPSMAP, where Subspace $\mathcal{P}$ and Subspace $\mathcal{Q}$ are formed from the LSP and patch features extracted from clean speech respectively. Once the CPSMAP is trained (using the procedure described in Section 4.4.2) we have a link between the spectrum of speech and the VT configuration responsible for its production.

Inference on CPSMAP is not as simple as inference on PSMAP. In the ABE problem the task was to estimate the spectrum that will be produced given a VT configuration. For the speech enhancement problem we need to move in the opposite direction during inference. The problem here is to estimate the VT configuration (specifically the LSP) that is responsible for a particular noisy spectrum of speech. Since the Subspace $\mathcal{P}$ is modeled with a HMM, the inference must result in a time series. The estimation problem can be succinctly described by Equation (52):

$$\max_{\mathbf{p}_{(1:t)}} p(\mathbf{p}_{(1:t)}, \rho_{(1:t)} | \mathbf{q}_{(1:t)}). \tag{52}$$

Simply stated, Equation (52) asks the following question "what is the most likely sequence of LSP ($\mathbf{p}_{(1:t)}$) that generates the observed sequence of patches ($\mathbf{q}_{(1:t)}$)?"

The problem (Equation (52)) can be split into two sub problems, as seen in the Equation (53) below:

$$\max_{\mathbf{p}_{(1:t)},\rho_{(1:t)}} p(\mathbf{p}_{(1:t)},\rho_{(1:t)}|\mathbf{q}_{(1:t)}) = \max_{\mathbf{p}_t} p(\mathbf{p}_t|\rho_t) \max_{\rho_{(1:t)}} p(\rho_{(1:t)}|\mathbf{q}_{(1:t)}). \tag{53}$$

Equation (53) suggests a two-step solution to the inference problem:

- **Problem 1** $\left[\max_{\rho_{(1:t)}} p(\rho_{(1:t)}|\mathbf{q}_{(1:t)})\right]$: Given the observed patch sequence ($\mathbf{q}_{(1:t)}$) what is the most likely VT sequence ($\rho^*_{(1:t)}$) responsible for its generation?
- **Problem 2** $\left[\max_{\mathbf{p}_t} p(\mathbf{p}_t|\rho_t)\right]$: Given a VT configuration (ρ_t) what is the most likely LSP ($\mathbf{p}_t$) that models the latent state?

4.6.1 Solution to Problem 1

Problem 1 can be solved using the Viterbi decoder. To see the solution clearly, we just need to rearrange some terms of the original problem. Equations (54) – (55) walk us through the steps to necessary to implement the Viterbi decoder:

$$\max_{\rho_{(1:t)}} p(\rho_{(1:t)}|\mathbf{q}_{(1:t)}) = \max_{\rho_{(1:t)}} p(\rho_{(1:t-1)},\rho_t = i,\mathbf{q}_{(1:t)}|\mathcal{M}) = \delta_t(i), \tag{54}$$

$$\delta_{t+1}(j) = \max_i \delta_t(i)\ a^{\rho}_{ij} \sum_{m=1}^{M} \mathbf{A}(m,\rho_{t+1}=i)\ p(\mathbf{q}_{t+1},\gamma_m). \tag{55}$$

4.6.2 Solution to Problem 2

Once we have traced the most likely state sequence (ρ^*) though the Subspace $\mathcal{P}$, the LSP coefficients for a given frame of speech can be estimated by maximizing the objective (Equation (56)):

$$\max_{\mathbf{p}_t} J(\mathbf{p}_t,\mathbf{q}_t) = \max_{\mathbf{p}_t} \sum_{\rho_t} p(\rho_t|\rho^*_{(t-1)}) p(\mathbf{p}_t|\rho_t)\Phi(\mathbf{q}_t), \tag{56}$$

where $\Phi(\mathbf{q}_t)$ is the $\sum_{n=1}^{M} p(\rho_t|\gamma_n)p(\mathbf{q}_t|\gamma_n)$.

Unfortunately, solution to problem described by the Equation (56) is not available in the closed-form. LSPs that maximize the objective J can be computed using a gradient ascent method. For the experiments presented in this chapter, we used resilient back-propagation [71] for gradient ascent. The gradient of the objective J required to implement resilient back-propagation is given by the Equation (57):

$$\frac{\partial J}{\partial \mathbf{p}_t} = -\sum_{n=1}^{N}\Big[p(\rho_t|\rho^*_{(t-1)})\Phi(\mathbf{q}_t)\sum_{l=1}^{L}\big[\big(w^{\rho}_{n,l}\big)\big(p_{n,l}(\mathbf{p}_t)\big)\frac{(\mathbf{p}_t-\mu^{n,l}_{\rho})}{\sigma^{n,l}_{\rho}}\big]\Big]. \tag{57}$$

An estimate of the speech spectrum can be computed once we have estimated the most-likely LSPs generating given sequence of noisy patches. The LSPs are first converted to LPC coefficients ($\bar{A}$) [68]. The LPC coefficients are then used to estimate the spectrum of clean speech (Equation (58)):

$$\widehat{X}_t = \frac{g^2}{P_{ss}(f)}, \tag{58}$$

where $P_{ss} = (|\bar{A}(\exp j2\pi f/f_s)|)^2$ and g is the linear prediction gain stored along with the LSP as feature vector for Subspace $\mathcal{P}$.

The *a priori* SNR for frame t is computed using the estimate of clean speech spectrum given by Equation (58), and the estimate of noise spectrum. At low SNR levels, however, there is a possibility that the estimated speech spectrum (Equation (58)) might not be accurate. Using such a degraded estimate of the spectrum will severely impact performance of the noise suppressor. Use of a correction term can mitigate this problem. The *a priori* SNR estimated from the *a posteriori* SNR ($\eta_t - 1$) of the current frame is used as a correction term. It prevents incorrect estimate of speech spectrum from severely impacting the gain of the noise suppressor. The complete CPSMAP-based *a priori* SNR estimator is given by Equation (59):

$$\widehat{\xi_t} = \alpha \cdot \Big(\frac{\widehat{X}^2_t}{\widehat{N}^2_t}\Big) + (1-\alpha)\cdot \max(\eta_t - 1, \xi_{min}). \tag{59}$$

The *a priori* SNR estimated using Equation (59) combines the *a priori* SNR

estimated from two source, a CPSMAP-based spectrum estimator and the *a posteriori* SNR estimate for the current frame. The parameter α in Equation (59) is just a weight. The parameter α in a DD *a priori* SNR estimator (Equation (40)) is the memory parameter that controls how many past frames impact the *a priori* SNR estimate of the current frame. In a CPSMAP-based *a priori* SNR estimator, parameter α prevents the degeneration of the estimator at low SNR conditions. Since the CPSMAP-based estimator does not rely on past results for making decisions for the current frame; it does not suffer from the lag suffered by a DD *a priori* SNR estimator.

4.6.3 Estimating *a priori* SNR using a CPSMAP

In this chapter, so far, we have presented CPSMAPs as model for speech production, an algorithm to train CPSMAP, and algorithm to infer *a priori* SNR using the CPSMAP. In the next section we will present the complete algorithm for denoising speech using a CPSMAP.

In the presence of noise, CPSMAPs trained on clean data are not useful. The patch features estimated from noisy speech no longer lie in the Subspace $\mathcal{Q}$; the noisy patches lie in the noisy Subspace $\bar{\mathcal{Q}}$.

To infer LSPs of clean speech using CPSMAPs we must first adapt Subspace $\mathcal{Q}$ to reflect the distortions caused by the operating environment. Subspace $\mathcal{Q}$ is adapted using the statistics of the estimated noise.

Using a voice activity detector we isolate and compute patch features for the noise frames. The noise is assumed to be stationary and Gaussian with a diagonal covariance. The noise is modeled as a Gaussian random variable with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$.

The patch features for the Subspace $\mathcal{Q}$ are computed using a series of linear operations on the magnitude spectrum; and latent states of Subspace $\mathcal{Q}$ are modeled

with Gaussians. These two factors make the adaptation of Subspace $\mathcal{Q}$ a simple linear operation. The means and variances of the adapted Gaussians of Subspace $\bar{\mathcal{Q}}$ are given by equations (60) and (61) respectively:

$$\boldsymbol{\mu}_{\gamma}^{m} = \boldsymbol{\mu}_{\gamma}^{m} + \boldsymbol{\mu}, \tag{60}$$

$$\boldsymbol{\sigma}_{\gamma}^{m} = \boldsymbol{\sigma}_{\gamma}^{m} + \boldsymbol{\sigma}, \quad \text{where} \quad 1 \leq m \leq M. \tag{61}$$

These adapted CPSMAPs are used to estimate *a priori* SNR. A noise suppression filter is calculated using the *a priori* SNR estimate. The complete procedure to enhance speech using CPSMAPs is given in Algorithm 2.

Algorithm 2 Denoise speech using the CPSMAP

1: Estimate the noise spectrum.
2: Compute patch features for noise and noisy speech.
3: Estimate noise statistics $(\boldsymbol{\mu}, \boldsymbol{\sigma})$
4: Adapt Subspace $\mathcal{Q}$ using Equations (60) and (61).
5: **while** Noisy Speech **do**
6: Update the noise statistics for the current frame.
7: Estimate the ML state sequence $\rho^{*}_{(1:t)}$ using Viterbi and Equation (55)
8: Estimate the VT parameters by solving Equation (56).
9: Compute the *a priori* SNR using Equations (58), (59) and noise estimate from Step 6.
10: Compute the suppression gain G.
11: Estimate the clean speech using the gain G generated in Step 10
12: **end while**

4.7 Experiments and Results

The performance of a PSMAP-based speech enhancement system was evaluated using four sets of experiments. Two different databases were used during this evaluation. A subset of utterances from 6 different speakers from the WSJ [66] was used for evaluation of perceptual quality of the enhanced speech. The 16 kHz WSJ audio was decimated to 8 kHz for perceptual quality experiments. Five to Six min of utterances from each speaker formed the training set. Three of the speakers in the dataset were males and the other three were females. Five kinds of noise distortions, babble,

pink, Volvo, white, and factory were selected from the NOISEX-92 database [81] to degrade speech during the testing phase. Speech enhancement tests were conducted on utterances not used in training of CPSMAPs.

The performance of the enhancement algorithm was also evaluated using an ASR system. The goal of such a test was to explore how the improvement in the estimate of *a priori* SNR impacts the accuracy of a traditional ASR setup. The ASR evaluation was performed on Aurora 2 dataset [51]. Aurora 2 consists of data degraded with additive noise and channel distortion. Three test sets provided with the task are contaminated with noise types seen in the training data (Set A), unseen in the training data (Set B), and additive noise plus channel distortion (Set C). This dataset provides a convenient baseline to evaluate and compare the performance of your ASR system to the standards and other results in the community.

The acoustic models for ASR were trained using the clean training utterances provided with Aurora 2. A standard "complex back-end" for Aurora 2 was trained using HTK [89]. The complex back-end consists of one HMM per digit treated as a whole word. Each HMM has 16 states per digit and 20 Gaussians per state. There is a three state silence model with 36 Gaussians per state and a one state short pause model tied to the middle state of silence. Standard 39 dimensional MFCC features consisting of 13 static, 13 delta, and 13 delta-delta features were used and C0 was used instead of log-energy [25]. The baseline word accuracy with no compensation on the test set is 63.38%.

4.7.1 System Configurations

We trained CPSMAPs using various model sizes, and we generated two different sets of CPSMAPs, one for Aurora 2 and another for WSJ.

Both the training and the test data were analyzed using 25 ms windows with 40% overlap between adjacent frames. 10 LSP coefficients augmented with the LPC gain

were used as observations of Subspace $\mathcal{P}$. The 2D-patch features described in Section 4.5.1 were used as the observation of Subspace $\mathcal{Q}$. Twenty three mel-filter banks were used to generate 23 patches for each frame of speech. The width of each patch was 9 frames. Only lower 9 DCT coefficients were retained for each patch. CPSMAPs of size 64, 128, 256, and 512 latent states per subspace were trained for both datasets. The GMMs modeling the latent states of Subspace $\mathcal{P}$ had 3 Gaussians per state. The CPSMAP training was carried out using techniques similar to the ones used for training simple PSMAP[1].

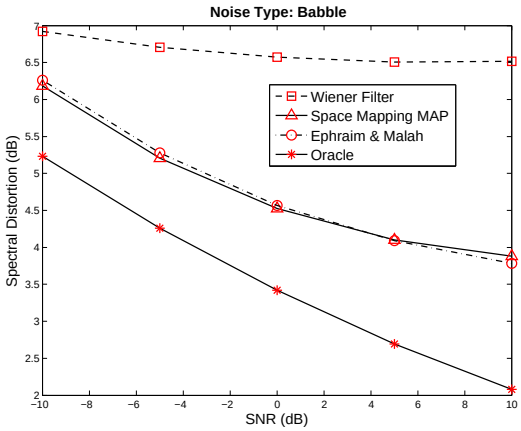
In all our experiments we used a simple energy tracking VAD for noise estimation. The VAD[2] used in our system is based on the VAD used in ETSI-AFE standard [26].

4.7.2 Experiment 1: CPSMAP-based Wiener vs. Oracle Wiener vs. E&M

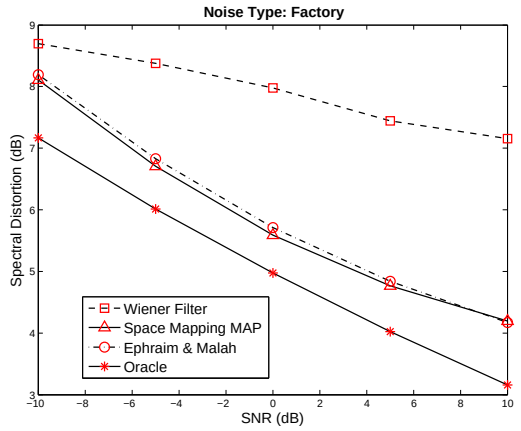
The experiments presented in this section were performed on the WSJ dataset using a STFT-based noise suppression system. The complete suppression system used in these experiments was very similar to the one described in the Figure 14. A CPSMAP with 128 latent states per subspace was used to perform the experiments presented in this section.

Figure 18 shows comparative performance for four speech enhancement techniques: Wiener Filter (WF) with a DD *a priori* SNR estimator, Wiener filter with *a priori* SNR estimated using CPSMAP (WPM), Ephraim and Malah speech enhancer (E&M) [23] and an oracle Wiener filter with perfect knowledge of the speech spectra (OWF). Under all types of noises, we observed that an OWF-based noise suppressor has the best objective performance and a WF-based system has the worst objective performance. The objective scores of an OWF provide an upper limit on the objective performance that can be achieved by a noise suppression system.

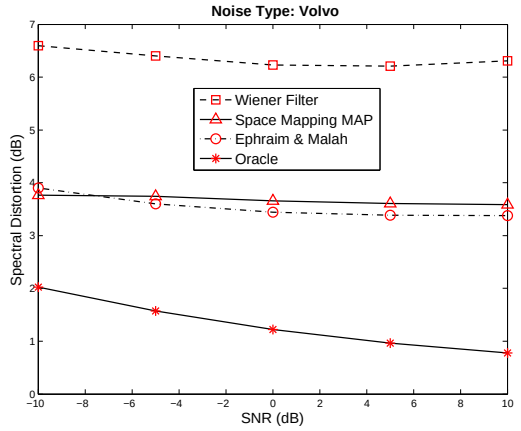
[1]Details description of the training strategy can be found in the Appendix A
[2]Appendix C presents the details of the VAD used in our system



(a) Babble Noise



(b) Factory Noise



(c) Volvo car Noise

Figure 18: Objective score comparison for Wiener, CPSMAP-Wiener, Ephraim and Malah, and Oracle Wiener filters.

Even though both a WF and an E&M noise suppressors use a DD *a priori* SNR estimator, an E&M noise suppressor outperforms a WF-based system because E&M uses Gaussian distribution to model the DFT coefficients, and E&M also uses log-MSE instead of MSE as a distortion measure. A WPM noise suppression system uses a CPSMAP-based model of speech production to improve the estimate of the *a priori* SNR. Thus, Wiener gain computed using this improved *a priori* SNR estimate elevates the objective performance of a WF-based system to match that of an E&M-based noise suppressor.

Both WPM and WF produce enhanced speech, but a WPM produces speech with better objective quality than a WF. A WPM-based system demonstrates similar trends for SD improvements for speech corrupted with factory, babble, and Volvo noises. A minimum of 1 dB improvement in the SD scores at low SNR conditions and a maximum of 3 dB improvement in SD scores at high SNR conditions is observed for all three kinds of noise degradations. These experiments (Figures 18(a) - 18(c)) also indicate that the CPSMAP-based *a priori* SNR estimator can work in variety of stationary and non-stationary noise conditions.

The results presented in this section are significant because they exhibit that substantial objective gains are possible by improving the estimate of *a priori* SNR. The CPSMAP-based *a priori* SNR estimator improves the objective performance of the WF to match that of an E&M noise suppressor.

4.7.3 Experiment 2: ASR Performance of Enhanced Speech

Aurora 2 dataset was used to perform experiments presented in this section. The noise suppression system used in this experiments were based on the system proposed in Section 4.2.3. The goal of this exercise was to evaluate the impact of noise suppression schemes on the word accuracies of an ASR system. This kind of ASR-scheme where noisy speech is first enhanced and then passed to the ASR is called a front-end

enhancement scheme. The ASR systems in front-end setup uses the acoustic models trained on clean speech to do recognition. Because we want a fair comparison we did not retrain the clean acoustic model. The word accuracy results for Aurora 2 are presented in Table 5.

Table 5: Average word accuracy for Wiener, E&M and CPSMAP (No Retraining)

SNR (dB)	**Average Word Accuracy**			
	Baseline	W-DD	E&M	CPSMAP-Wiener
20	96.11	54.21	83.81	95.65
15	88.91	43.31	77.40	92.86
10	72.02	31.83	67.94	85.41
5	43.00	20.64	51.22	66.91
0	16.89	12.29	28.49	35.63
Average	63.38	32.45	61.77	75.29

The DD Wiener filter takes a very big hit in word accuracy (32.45%) if the acoustic models are not retrained, whereas the CPSMAP-based Wiener filter (75.29%) does not suffer any loss in performance. The CPSMAP enhanced speech actually improves word accuracy of the baseline acoustic models. The E&M based front-end scheme also suffers a small loss in word accuracy.

Not surprisingly, the accuracies of all these systems can be improved by simply retraining the acoustic models. On retraining the said models we observe the performance of W-DD and E&M systems tends to be slightly better than that of the baseline system but never close to the accuracy of the CPSMAP-based Wiener filter.

Looking at the results from experiments 1 and 2 we can state the CPSMAP-based front-end improves the perceptual performance of processed speech without negatively impacting the accuracies of the ASR system.

4.7.4 Experiment 3: mel-warped Wiener vs. CPSMAP based mel-warped Wiener

All the experiments presented in this section are based on the ETSI-AFE [26]. The AFE is a standard front-end proposed by ETSI that uses a two-stage Wiener filter to

suppress noise. The Wiener filter used in this system is designed in the mel-domain.

The experiments presented in this section were performed on the WSJ dataset. Noise of type babble, factory, pink, Volvo, and white from the Noisex database was used to degrade the utterances in the test set.

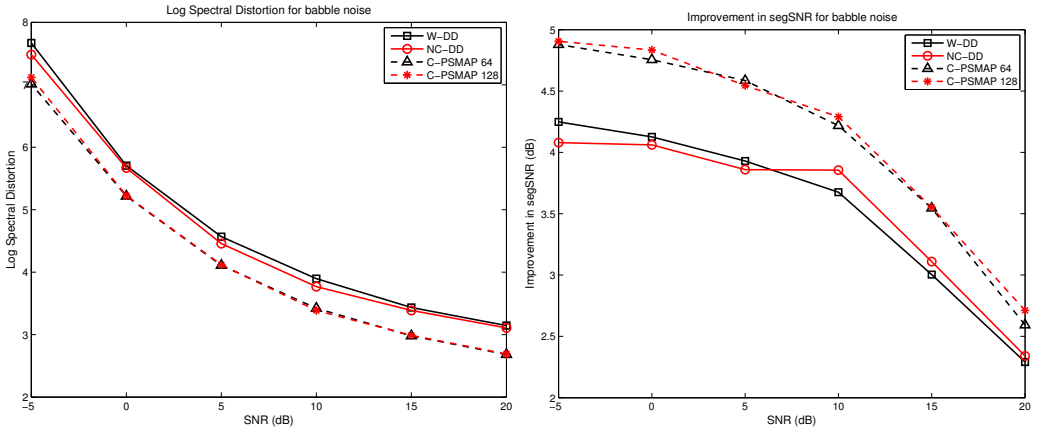
We designed three front-end systems using ETSI-AEF. The three system compared in this experiment were designed by replacing the DD *a priori* SNR estimator in the standard ETSI-AFE system by other *a priori* SNR estimators. Three systems compared in these experiments are:

- An AFE system with standard DD *a priori* SNR estimator.
- An AFE system with non-causal (NC) DD *a priori* SNR estimator.
- An AFE system with a CPSMAP-based *a priori* SNR estimator.

The performance of these three systems was compared using SD and segSNR as metrics. Figures 19(a) - 20(d) plot the objective performance of the system at SNR ranging from -5 dB to 20 dB.

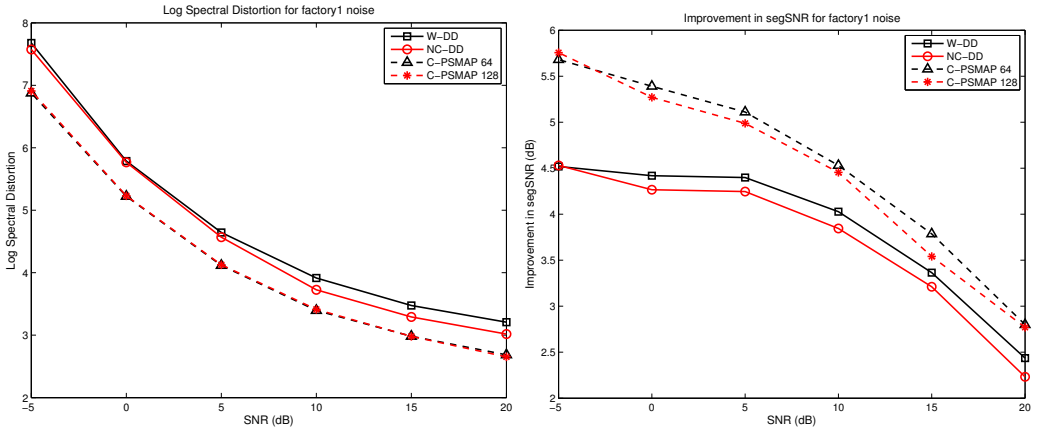
We performed experiments using CPSMAP of various sizes. In this results section we have plotted objective scores for two different systems: a system with small CPSMAP that has 64 latent states per subspace, and a system with large CPSMAP that has 128 latent states per subspace. We observed that there is a small difference in the performance of large and small CPSMAPs.

During the experiments we discovered, that a CPSMAP of size less than 64 had performance similar to a DD-based AFE. The AFE systems with large CPSMAPs (size greater than 128) displayed a very small improvement in performance over the AFE systems using CPSMAP with 128 states per subspace (see Figures 19(a) - 20(d)). Based on these experiments we concluded that CPSMAP of size 128 is a good tradeoff between objective improvement and computational cost.



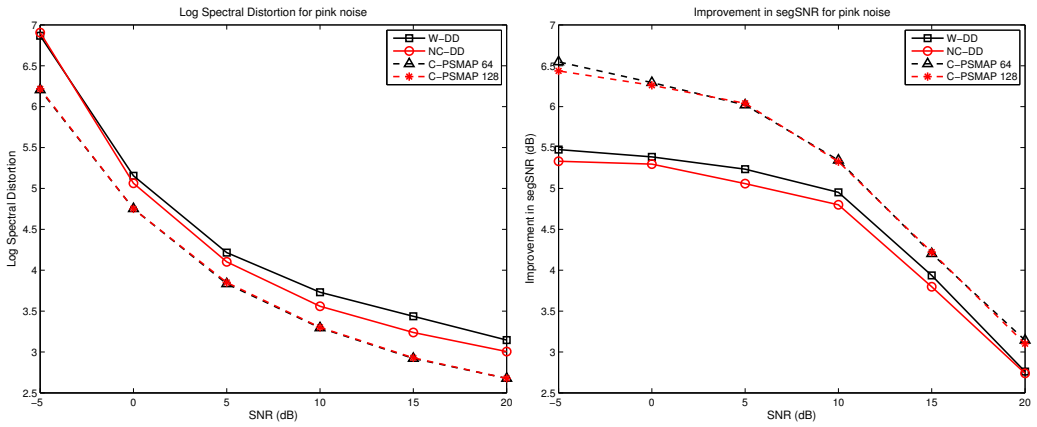
(a) Babble SD

(b) Babble segSNR



(c) Factory SD

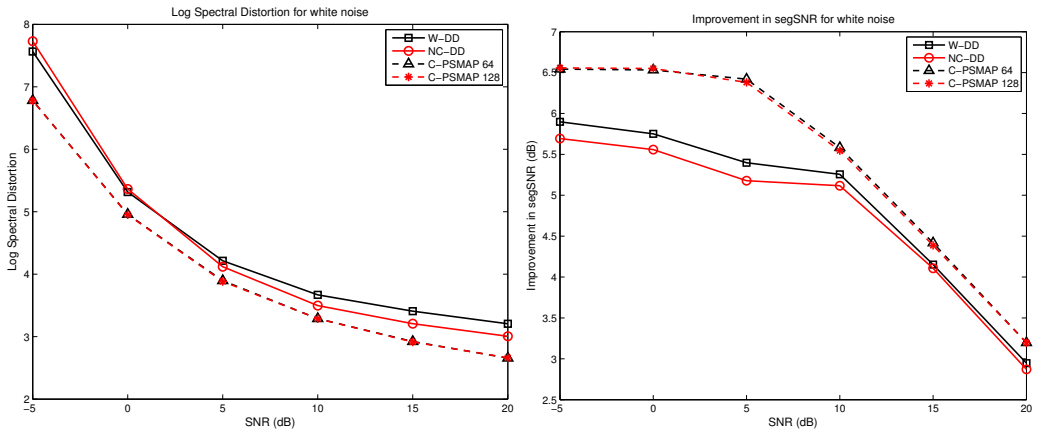
(d) Factory segSNR



(e) Pink SD

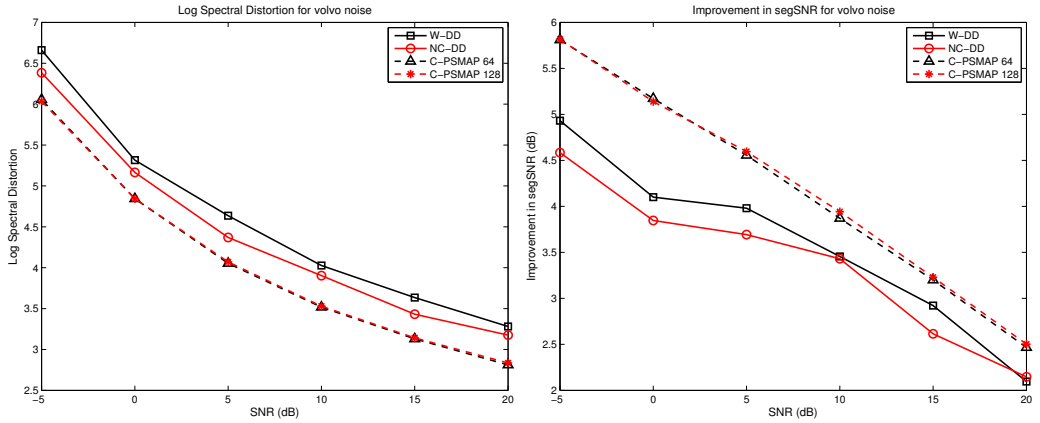
(f) Pink segSNR

Figure 19: Objective score comparisons for DD-Wiener, NC-Wiener and CPSMAP-Wiener (Babble, Factory, and Pink noise).



(a) White SD

(b) White segSNR



(c) Volvo SD

(d) Volvo segSNR

Figure 20: Objective score comparisons for DD-Wiener, NC-Wiener and CPSMAP-Wiener (White and Volvo noise).

A CPSMAP-based WF outperforms both the non-causal DD-AFE and the DD-AFE systems. The performance improvements that CPSMAP demonstrate are significant. In most cases, both the SD and segSNR improvement are at least 1 dB. But for noisy environment, where the noise is of the type babble/crosstalk (Figure 19(b)), we observe that CPSMAP have a greater advantage over the standard systems especially at low SNR conditions. A CPSMAP uses a speech production model to estimate the spectrum of clean speech, therefore, the estimates of the spectrum generated using a CPSMAP closely represent human speech spectra even in the cases where noises (babble) have statistics similar to those of speech.

The use of NC-DD estimator does provide some improvements over the standard DD-AFE, but the improvements are not as significant as those provided by the CPSMAP-based *a priori* SNR estimator.

4.7.5 Experiment 4: ASR Performance for mel-warped System

The next sets of experiments were performed using Aurora 2 models and data. ETSI-AFE-based front-end was used to enhance speech[3]. A CPSMAP with 128 latent states per subspace was used for all the experiments presented in this section.

Figure 21 shows the relative improvement in word accuracy of AFE if we use CPSMAP to compute *a priori* SNR. The CPSMAP-based front-end shows an average relative improvement[4] of 12.82% over the baseline AFE system. A CPSMAP-based AFE system performs better than the standard AFE system at all SNR levels. The gains in performance are smaller at lower SNR levels of 0 dB and 5 dB. We observe higher relative improvement in accuracy at SNR of 10 and 15 dB. The CPSMAP-AFE results in over 35% and 45% relative improvements in accuracies at SNRs of 10 dB and 15 dB. This result is especially encouraging, as most of the commercial systems (e.g., GSM) usually tend to operate in 10-15 dB SNR region.

Tables 6 — 8 compare the word recognition accuracy for ETSI-AFE based on DD *a priori* SNR estimator (AFE) and the ETSI-AFE using a CPSMAP-based *a priori* SNR estimator (CP). The CPSMAP-based AFE has better word accuracy than the standard AFE for all the noise and channel conditions. One interesting point to note is the performance of the system in babble noise. Both the front-ends have trouble enhancing speech degraded with babble noise, this is due to the fact that babble noise has statistical properties close to that of the signal of interest (speech).

[3]Details about the implementation of CPSMAP-based front-end used for these experiments can be found in Appendix C

[4]Average relative improvement in word accuracy is calculated by comparing absolute improvement to the failure rate: $\left(\frac{WA_1 - WA_2}{100 - WA_2}\right)$.

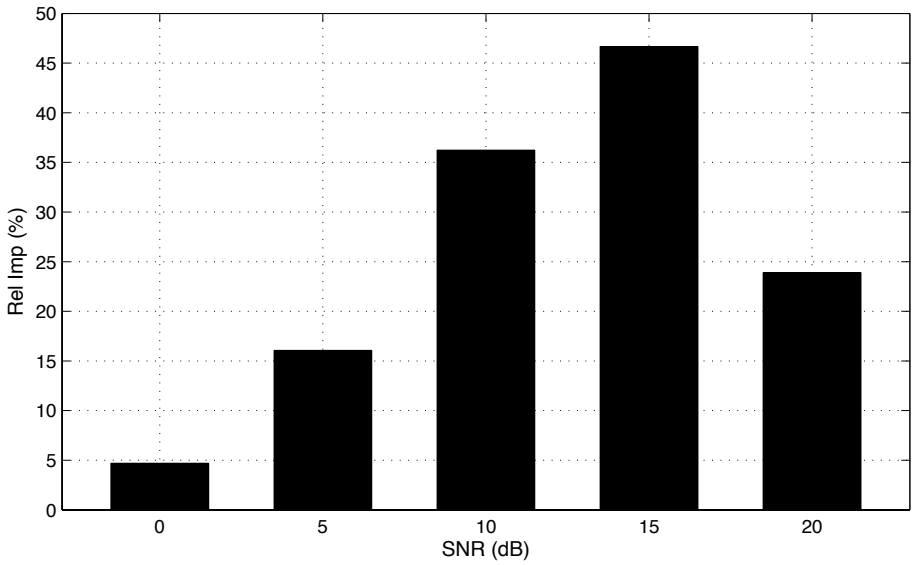


Figure 21: Relative improvement in word accuracy of CPSMAP-AFE over ETSI-AFE.

This similarity makes it difficult for any statistical modeling technique to separate speech from background noise. In the presence of babble noise, the CPSMAP-based AFE (84.51%) shows slight improvement in word accuracy over the standard AFE system (84.06%).

Table 6: Aurora 2 Set A word accuracy comparisons (CP is the CPSMAP-based ETSI-AFE)

SNR (dB)	**Subway**		**Babble**		**Car**		**Exhibition**	
	AFE	CP	AFE	CP	AFE	CP	AFE	CP
20	99.05	99.05	98.73	99.34	99.14	99.92	98.92	98.92
15	97.76	97.76	97.13	97.83	98.36	99.79	98.12	98.12
10	93.80	97.36	92.90	92.60	96.60	98.72	94.94	98.76
5	85.66	88.21	80.02	78.81	90.13	93.02	86.67	91.39
0	65.92	64.29	51.54	53.98	71.67	74.09	65.94	71.03
Average	88.44	89.33	84.06	84.51	91.18	93.11	88.92	91.64

Table 7: Aurora 2 Set B word accuracy comparisons

SNR (dB)	Restaurant		Street		Airport		Station	
	AFE	CP	AFE	CP	AFE	CP	AFE	CP
20	98.96	98.96	98.67	99.34	99.05	99.71	99.17	99.17
15	96.81	97.73	96.49	99.03	96.33	99.28	97.09	99.75
10	92.42	93.06	92.68	96.13	93.23	95.23	94.50	97.60
5	79.89	80.35	81.95	86.46	84.21	86.75	84.26	88.34
0	53.48	55.19	61.54	62.27	62.00	63.39	62.73	68.78
Average	84.31	85.06	86.27	88.65	86.96	88.87	87.55	90.73

Table 8: Aurora 2 Set C word accuracy comparisons

SNR (dB)	Subway M		Street M	
	AFE	CP	AFE	CP
20	98.50	98.49	98.52	98.46
15	97.76	98.80	97.13	97.52
10	91.58	95.15	91.62	93.47
5	81.45	86.91	82.10	82.32
0	62.82	63.49	61.03	60.11
Average	86.42	88.57	86.08	86.38

4.7.6 Perceptual Tests

Objective tests lack the ability to quantify glitches and perceptual artifacts in synthesized utterances. To measure the perceptual quality, subjective tests were performed on WSJ dataset. The tests employed 25 listeners. All the tests were performed using the same setup. All of the subjects used a Sony MDR-V600 circumaural headphone. The user were encouraged to set the volume at the start of the experiment. All the listeners used the same set of baseline utterances to set the volume.

To measure the quality of the enhanced speech, we conducted mean opinion score (MOS) [17, 1]. Table 9 shows the score criteria set for tests.

The subjective tests were performed at two SNR conditions 0 dB and 10 dB. Table 10 documents the results of the MOS tests for three different algorithms: E&M, ETSI-AFE and ETSI-AFE with CPSMAP.

At SNR of 10 dB the quality of unprocessed speech itself is 'fair'. Any artifacts

Table 9: Perceptual test scoring criteria

Score	Impairment
5 (Excellent)	Imperceptible
4 (Good)	(Just) Perceptible but not Annoying
3 (Fair)	(Perceptible and) Slightly Annoying
2 (Poor)	Annoying (but not Objectionable)
1 (Bad)	Very Annoying (Objectionable)

Table 10: Subjective test scores of noise suppression

SNR (dB)	Noisy Speech	Cleaned Speech		
		E&M	AFE-DD	CPSMAP
0	1.6	2.4	2.6	2.8
10	3.4	3.6	3.6	3.8

introduced during processing of high SNR speech will impact MOS scores negatively; therefore, it is challenging to improve MOS scores of speech at high SNR. Both the E&M and ETSI-AFE system improve the subjective quality by suppressing noise. Both these systems produce an improvement of 0.2 in MOS score of enhanced speech. At high SNR level the E&M gain function behaves similar to the Wiener gain function [22], that is why both E&M and AFE systems have similar performance at high SNR. The CSMAP enhanced speech has an improvement 0.4 MOS points over the unprocessed noisy speech.

At low SNR levels, it is relatively easier to improve the subjective quality of enhanced speech. According to the listeners the quality of unprocessed speech at 0 dB SNR was poor (MOS score of 1.6). An E&M noise suppressor improved the subjective quality and boosted the MOS score of processed speech to 2.4. The DD-AFE system is a two-stage Wiener filter; consequently it performs better than the single-stage E&M system. The speech processed using DD-AFE resulted in MOS scores of 2.6. The best MOS score of 2.8 was observed for speech processed using a CPSMAP-based AFE.

A CPSMAP models the process of speech production. A suppression system based

on CPSMAP therefore produces *a priori* SNR estimates that closely represent the true speech spectrum. The improved fidelity of the *a priori* SNR estimator results in a suppression filter that produces less audible artifacts in synthesized speech, which results in higher listener satisfaction and higher MOS scores.

4.8 Conclusions

In this chapter we presented a new statistical model that probabilistically maps subspaces and transforms between subspaces. This model also imposes constraints on state transition within a subspace by using a HMM to model the latent states of the subspace. The CPSMAP based model for speech captures the temporal dynamics of the speech production process better than the PSMAP based model, and hence represent the speech production process more faithfully.

This new model is applied to the problem of speech enhancement. We performed a battery of subjective and objective tests to compare performance of a CPSMAP-based noise suppressor to Ephraim and Malah's, and ETSI-AFT systems. The algorithm suggested in this chapter not only obtains better objective scores than the existing systems, but it also performs better on subjective tests.

The ETSI-AFE based front-end is one of the best feature enhancers for the ASR systems. Addition of CPSMAP to the *a priori* SNR estimator of ETSI-AFE improves the system. This new CPSMAP-based front-end (88.68%) has better word accuracy on Aurora 2 than the standard ETSI-AFE system (87.02%) and the baseline VTS system (88.27%).

CHAPTER V

ACOUSTIC MODEL ADAPTATION

5.1 Introduction

Noise is a major culprit in the poor performance of automatic speech recognition systems. Speech recognizers often fail due to the mismatch in the training and deployment conditions (channel and noise effects). Despite years of research, automatic speech recognition (ASR) in noisy environments remains a challenging problem since there are many possible types of environmental distortion, and it is difficult to compensate for all of these distortions accurately.

As suggested in the previous chapter, one can use front-end enhancement schemes to clean up the noisy features so that they match the clean features that were used to train the acoustic model. The front-end enhancement schemes are typically simpler and computationally efficient than their back-end counterparts. The front-end methods have shown improved performance on several tasks, they all, by definition, make point-estimates of the clean speech features. Errors in these estimates can cause further mismatch between the features and the acoustic model, resulting in degraded performance.

Model adaptation techniques avoid this problem by compensating the probability distribution of the acoustic model directly. Model adaptation techniques fall in two categories: data driven, and nonlinearity-based.

Some adaptation schemes such as MLLR [54] and MAP adaptation [35], are data-driven methods that do not make any assumption about the noise or channel. These schemes require a moderate amount of adaptation data to learn adaptation parameters. In situations where there is limited adaptation data, reduced-parameter methods

such as CMLLR [21, 32] have proven to be useful.

The second set of model adaptation techniques exploit the known nonlinear relationship between cepstra of clean speech, noisy speech, and noise [33]. In vector Taylor series (VTS) [62] adaptation, the nonlinear function is linearized using a Taylor series. In [42], an unscented transform was used to estimate noisy speech distribution using a small number of speech and noise sample points. In [52] a linear spline was used to map the nonlinearity, and the phase variations around the spline were modeled with a segmental variance for each spline section.

In this chapter, we propose a novel method based on PSMAPs [50, 49] to adapt the Gaussians of acoustic HMMs to the noisy environment. As demonstrated Chapter 4, PSMAPs can be used at the front-end of the recognizer to clean noisy speech. The cleaned speech is passed to the ASR system for recognition. Cleaning the utterance does indeed improve the accuracy of the ASR system, but the improvements are limited. Using noise suppressor at the front-end of the recognizer mandates retraining of acoustic models trained on clean speech database. Use of PSMAPs at the back-end of the ASR system avoids retraining the acoustic models. Further use of back-end schemes for model adaptation leads to systems that have better accuracy than front-end ASR systems.

There are several differences between the existing adaptation schemes and PSMAP-based proposed method. First, the proposed algorithm automatically extracts the nonlinear relationship between the parameters (clean speech, noisy speech and noise) using the training data. Second, the proposed PSMAP-based model adaptation scheme does not approximate the nonlinear relationship between dynamic coefficients as is commonly done in existing (VTS, PMC, LSI) adaptation schemes.

The next section will present the background information on existing model adaptation techniques, and highlight the problems and sources of improvement in those schemes.

5.2 Nonlinear Distortion and Model Adaptation

5.2.1 Nonlinear Distortion of Speech Cepstra

The power spectrum of noisy speech (Y) can be expressed as the function of the power spectrum of clean speech (X) and noise (N) as shown in the equation below:

$$|Y|^2 = |X|^2 + |N|^2 + 2|X| \cdot |N| cos(\phi), \tag{62}$$

where ϕ is the relative phase between the speech and the noise. MFCCs for the frame of speech are extracted from the power spectrum by computing the DCT of the log of the output of each mel-filter bank. Using this knowledge, we can exactly calculate relationship between the MFCCs clean speech, noisy speech, and noise.

Let y, x and n represent the cepstra of noisy speech, clean speech, and noise respectively. If C is the DCT matrix and D is the pseudo inverse of matrix DCT matrix, then $x = CX$, $y = CY$ and $n = CN$ are the MFCC coefficients of clean speech, noisy speech, and noise respectively. Using this notation, the relationship between the cepstra of clean speech, noisy speech, and noise is given by the Equation (63):

$$y = n + C \log(1 + e^{D(x-n)} + 2\alpha e^{D(x-n)/2}), \tag{63}$$

$$y - n = C \log(1 + e^{D(x-n)} + 2\alpha e^{D(x-n)/2}), \tag{64}$$

where α represents the contribution of the phase term ϕ [20].

Following the convention suggested by McAulay and Malpass,'$u = x - n$' is the *a priori* SNR and '$v = y - n$' is *a posteriori* SNR[1]. Using this notation, we can rewrite the Equation (64) as a function of *a priori* and *a posteriori* SNRs:

$$v = C \log(1 + e^{Du} + 2\alpha e^{Du/2}). \tag{65}$$

[1] x, n, and y are all log cepstras, so $v = y - n$ is actually $v = \log \frac{e^y}{e^n}$

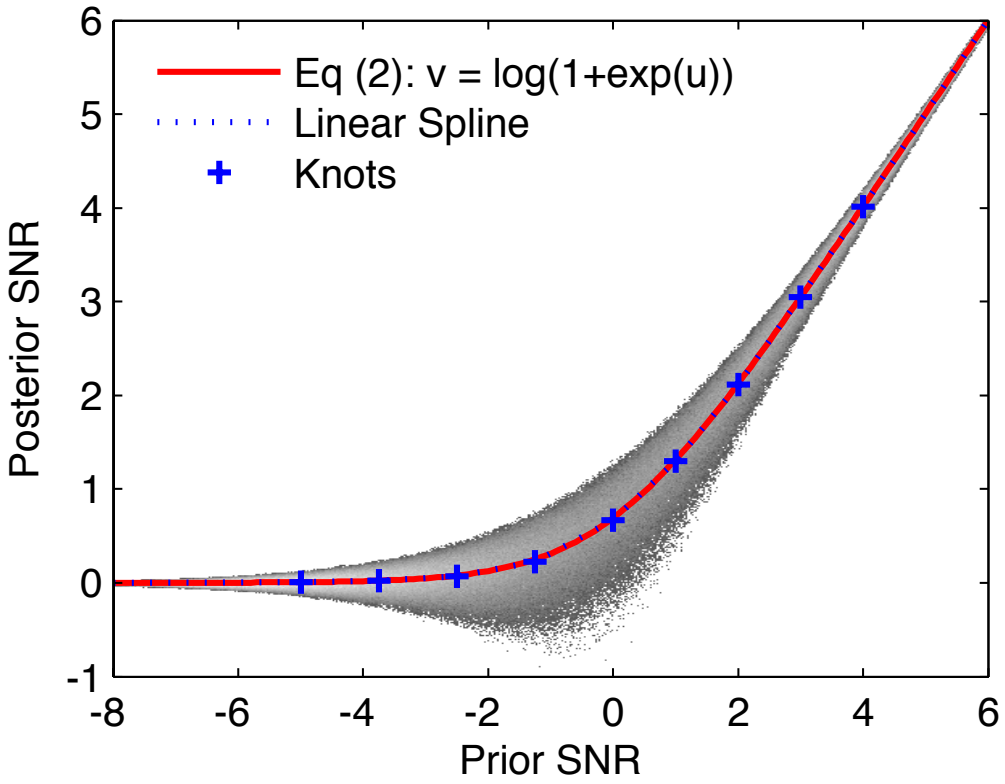


Figure 22: Plot of $x-n$ vs. $y-n$, showing the scatter of the true data and the mode of the nonlinear relationship ($v = log(1 + exp(u))$).

Some of the most successful model adaptation algorithms [33, 56, 55, 74, 52] have used a modified form of the Equation (65), this modification is implemented to simplify the linearization of the Equation (65). The phase term (α) in the Equation (65) is dropped before linearization to yield:

$$v = C \log(1 + e^{Du}) \tag{66}$$

Figure 22 is the scatter plot generated using cepstral data from Aurora 2. The data from the 16^{th} cepstral coefficient was used to generate the scatter plot. The Red solid line (mode of data) in Figure 22 is the plot of Equation (66).

VTS algorithm suggested by Moreno and colleagues [62] linearizes Equation (66) using a vector Taylor series expansion [2]. In doing the linearization, VTS ignores the correlation between y and n. VTS also ignores the variance of cepstra around the

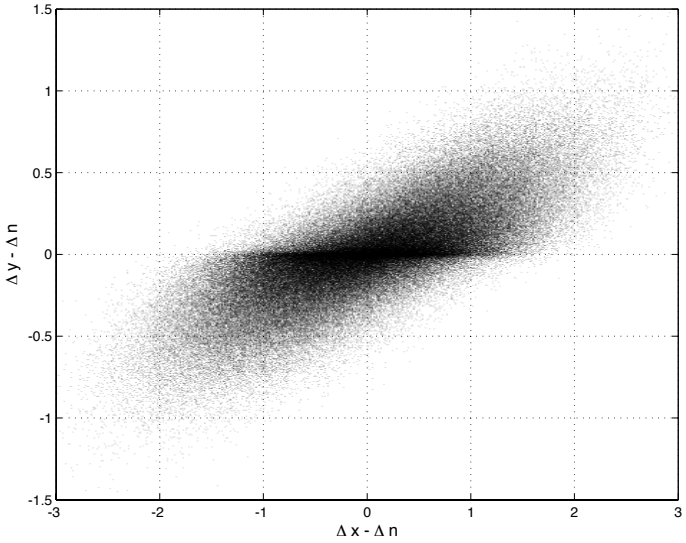
mode line. Even with these simplifications VTS-based ASR systems outperform the ETSI-based ASR system (see Table 17).

Deng and colleagues in [55] suggested a variant of the standard VTS algorithm that models both the mode and variance of the data around the mode. To model the variance of data around the mode [55] precomputes the values of α from the training data and stores it in a codebook. This modeling of variance leads to significant improvement over the VTS system. The best results, however, are obtained for a static value of $\alpha = 2.5$. The value of parameter $\alpha = 2.5$ is specific for Aurora 2 and was heuristically determine.

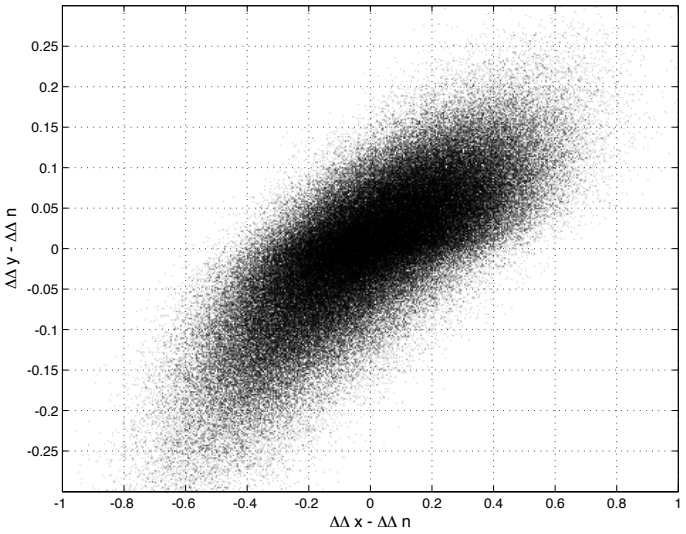
Kalgaonkar and colleagues [52, 74] used a spline for linearization (LSI) of Equation (65). The variance around the mode was captured using a zero mean Gaussian for each segment of the spline. The acoustic models adapted with linear spline have better word accuracy than the models adapted using VTS, however, word accuracies of these systems do not surpass the accuracies of the systems suggested by Deng [55] ($\alpha = 2.5$).

In all the variations of VTS/LSI, the log nonlinearity is approximated with a linear transform $\mathbf{F}$. The transform $\mathbf{F}$ is estimated for each Gaussian of the acoustic model. For a large acoustic model, this adaptation process is computationally expensive. Further, for such an expensive process we only get the first order approximation of the *log* function. PSMAP-based adaptation system will learn the nonlinear transform for training data, thereby eliminating the need for approximate linearization.

Figures 23(a) and 23(b) show the scatter plots of delta and delta-delta coefficients respectively. The relationship between the dynamic coefficients of clean speech, noisy speech and noise is not the same as that of the static coefficients. This can be seen by comparing plots for static coefficient (Figure 22) and dynamic coefficient (Figure 23). The relationship between the dynamic coefficients is not as simple to estimate



(a) Delta Coefficient



(b) Acceleration Coefficient

Figure 23: Scatter plots for $x - n$ vs. $y - n$ for dynamic coefficients.

or linearize as that of the static coefficients (compare Figures 22 and 23). The primary cause for the discrepancy between models of static and dynamic parameters is the regression function used to compute the dynamic coefficients [89]. To adapt the parameters of dynamic coefficients, algorithms such as VTS, LSI, etc., [62, 52, 42, 37] use a continuous time approximation of derivatives to simplify the relationship between the cepstras of the dynamic coefficients. This approximation results in relaxing

the modeling of dynamic coefficients and facilitates the use of static-coefficient's linearization function $\boldsymbol{F}$ to adapt the parameters of dynamic coefficients.

Unlike the VTS/LSI based systems, a PSMAP-based adaptation scheme does not require explicit knowledge of the relationship between the cepstras of clean speech and noisy speech. PSMAPs will learn the nonlinear transform between the cepstras for the static as well as dynamic coefficients using training data. This property of PSMAP will avoid the unnecessary linearization or approximation of the transform.

In the next section, we will present a PSMAP-based acoustic model adaptation system that does not directly approximate the mode of Equations (65) or (66) but rather tries to capture the relationship between the distributions of two probability spaces ([x;n] and y).

5.3 Model Adaptation using Probabilistic Space Maps

PSMAPs provide a latent variable framework to map two subspaces. Using exemplar data, PSMAP can learn the transform $\Psi()$ between the variables that constitute the subspaces. We can exploit this property of PSMAPs and learn the nonlinear transform (Equation (63)) between cepstra of clean speech, noisy speech, and noise. To train such a PSMAP we need access to stereo data. Aurora 2 [39] has a multi-condition training data set, with roughly 8000 clean utterances and 8000 noisy utterances. The SNR for noisy speech in the training set varies from 5 dB to 20 dB.

The observations ($\mathbf{p}$) of the Subspace $\mathcal{P}$ are the cepstra for clean speech and noise $[\mathbf{x}^T,\ \mathbf{n}^T]^T$, and the observations ($\mathbf{q}$) of the Subspace $\mathcal{Q}$ are the cepstra of the noisy speech $[y]$. The parameters of the model $\mathcal{M} = (\boldsymbol{\rho}, \boldsymbol{\gamma}, \boldsymbol{A})$ are learned from the training data using the method described in the Section 2.3.

A trained PSMAP is a collection of functions (ρ_m, γ_n) that makeup the basis of the distribution of respective subspaces. The second component of a PSMAP is a the transition matrix $\boldsymbol{A}$, that encodes the transform between the basis of the Subspace

$\mathcal{P}$ and Subspace $\mathcal{Q}$.

During a live ASR run, the goal is to estimate the noisy acoustic model given the clean acoustic model and the estimate of noise. We assume the noise to be normally distributed. Each Gaussian of the noisy acoustic model ($g_y = \mathcal{N}(\mu_y, \Sigma_y)$) is estimated from the equivalent Gaussian of the clean model ($g_x = \mathcal{N}(\mu_x, \Sigma_x)$) and the noise ($g_n = \mathcal{N}(\mu_n, \Sigma_n)$). First step in adaptation is to represent vector of clean (from the acoustic model) and noise Gaussian $g = [g_x; g_n]$ as a linear combination of the basis of Subspace $\mathcal{P}$ (i.e. $g = \sum_{i=1}^{N} F_i \cdot \rho_i$). In the second step we use the transition matrix ($\boldsymbol{A} \in \mathbb{R}^{M \times N}$) to estimate the weight for basis of Subspace $\mathcal{Q}$ ($\boldsymbol{\gamma}$). Combining the weights with the bases of Subspace $\mathcal{Q}$ will yield the adapted distribution. The adaption process can be succinctly represented using Equation (67):

$$p(y) = \sum_{m=1}^{M} \Big(\boldsymbol{A} \cdot F\Big)\mathcal{N}(\boldsymbol{\mu}_\gamma^m, \boldsymbol{\sigma}_\gamma^m), \tag{67}$$

where $F = [F_1, F_2, \ldots, F_N]^T$ is a vector of weights of basis of Subspace $\mathcal{P}$.

Computing the weights 'F' is a iterative procedure, which can get really expensive for large acoustic models. The Gaussian distribution is a member of the exponential family, therefore, it can be completely described by its sufficient statistics. This property of the normal distribution can be exploited to avoid the explicit solution of the linear weight function F.

Julier and Uhlmann [46] suggested the use of *sigma-points* as an elegant solution for this problem. Sigma-points for a distribution $p(x)$, where ($x \in \mathbb{R}^L$) are a set of deterministically chosen ($2L+1$) points and associated weights (w). The weights can be positive or negative but must always sum to unity.

For a Gaussian distribution, the sigma-points must completely capture the first and the second moments. One such set of sigma-points for a Gaussian distribution with mean $\boldsymbol{\mu}$ and Covariance $\mathbf{R}$ are given by Equation (68) - (71)

$$u^{(0)} = \mu \tag{68}$$

$$w^{(0)} = 1 - \frac{L}{3} \tag{69}$$

$$u^{(k),(k+L)} = \mu \pm \left(\sqrt{\frac{L}{1-w^{(0)}}\mathbf{R}}\right) \tag{70}$$

$$w^{(k)} = w^{(k+L)} = \frac{1-w^{(0)}}{2L} \tag{71}$$

The complete procedure for acoustic model adaptation using PSMAP and sigma-points is described in Algorithm 3:

Algorithm 3 Adapting the clean acoustic HMM for Noisy Conditions

1: Estimate the noise statistics (μ_n, Σ_n) for an utterance.
2: **for** Each Gaussian $(\mathcal{N}(\mu_x, \Sigma_x))$ in the clean acoustic model **do**
3: Compute the sigma-points for distribution $p(u)$ using Equations (68) - (71).

- $p(u) = \mathcal{N}\Big([\mu_x; \mu_n],\ daig([diag(\Sigma_x); diag(\Sigma_n)])\Big)$

4: Estimate $y^{(k)}$ for each sigma-point $(u^{(k)})$ using Equation (10).
5: Estimate the mean and variance adapted (noisy) Gaussian $\mathcal{N}(\mu_y, \Sigma_y)$ as follows:

- $\mu_y = \sum_{k=1}^{2D+1} w^k y^k$
- $\Sigma_y = \sum_{k=1}^{2D+1} w^k (y^k - \mu_y)(y^k - \mu_y)^T$

6: **end for**

where '**diag**' is the diagonal operator for a matrix, which converts a full matrix to a vector of diagonal elements. This operator also converts a vector to a full matrix where every element but the diagonal is zero.

In the presence of noise the Gaussians of the clean acoustic model are adapted using the statistics of the noise. Recognition can be performed once all the Gaussians of the acoustic models are adapted. During the training phase, three separate PSMAPs are generated (one for each static, delta, and delta-delta coefficients). During the

adaptation phase all Gaussians of the acoustic model are adapted using the process described in Algorithm 3. Each set of coefficient is adapted using its own PSMAP. The newly adapted HMM is used for ASR.

5.4 Experiments and Results

The effectiveness of the new algorithm was evaluated by conducting a series of experiments on the Aurora 2 connected digit recognition corpora. The acoustic models were trained using HTK speech recognition system [89]. Aurora 2 consists of artificially degraded data. Eight different kinds of noise with SNR varying from 0 dB to 20 dB were added to clean speech to generate the test data. Aurora 2 consists of test set that is divided into three categories: Set A is degraded with noise types seen in the training data, Set B is degraded with noise that is not present in the multi-conditioned training data and Set C contains both additive noise and channel distortion.

Standard complex acoustic models were trained from the clean training utterances. A HMM with 16 states per digit and 20 Gaussians per state was created for each digit as a single word. In addition, there is a three state silence model with 36 Gaussians per state and a one state short pause model, which is tied to the middle stage of silence model. Standard 39 dimensional MFCC features consisting of 13 cepstral features plus 13 deltas and 13 delta-delta's features were used in the experiments. The cepstral coefficient of order zero (C0) is used instead of log energy [25]. The distribution of the cepstral coefficients of noise is assumed to be Gaussian with diagonal covariance. The first ten frames of each utterance were used to estimate the mean and covariance of the noise for that utterance. The PSMAPs were trained with 64 Gaussians in each subspace.

We performed acoustic model adaptation experiments using PSMAP of sizes 32, 64, 128, and 256 latent states per subspace. We observed that acoustic model adapted

using a PSMAP of size greater than 64 produce same word accuracy as that of the acoustic model adapted using a PSMAP of size 64. An acoustic model adapted using PSMAP of size 32 has word accuracy (70.01%), comparable to the baseline acoustic model trained on clean speech data. We also observed that imposing sparsity on larger PSMAP (128 and 256 latent states) did not result in any improvement, hence all the results presented in the next section use a PSMAP with 64 latent states per subspace.

The VTS and LSI schemes adapt the means of static, and dynamic coefficients but only adapt the variance of the static and delta coefficients. The VTS/LSI schemes actually observe degradation in word accuracy if the variance of delta-delta coefficients is adapted for noise. A PSMAP-based model adaptation scheme adapts means and variances of all the coefficient of the acoustic model.

Tables 11 – 13 shows the % word accuracy results for the adapted models for Aurora 2. As expected the presence of babble noise (89.38%) and restaurant noise (89.19%) significantly impacts the accuracy of the system. This is due to the similarity between the characteristic of the speech and the noise spectrum. The PSMAP-based adaptation scheme does not account for the presence of channel, therefore, the average accuracy for set of utterances degraded with subway noise and channel (92.92%) is worse that average accuracy of the utterances degraded with only subway noise (93.23%).

Tables 14 – 16 compares the word accuracy of models adapted with 3 different schemes: VTS [62], LSI [52] and PSMAP. The PSMAP-based model adaptation outperforms both VTS and LSI schemes. The acoustic models adapted using PSMAP have better accuracy than the models adapted using VTS and LSI for all types of noises and channel distortions, at all SNR levels. The average relative improvement of PSMAP over VTS is 26.65% compared 19.69% achieved by LSI. This result is

Table 11: Aurora 2 word accuracy using PSMAP for Set A

SNR (dB)	**Set A**			
	Subway	Babble	Car	Exhibition
∞	99.6	99.6	99.6	99.6
20 dB	99.17	98.34	99.16	98.58
15 dB	98.43	97.19	98.45	97.62
10 dB	96.78	94.59	96.69	94.14
5 dB	92.91	88.51	92.45	87.23
0 dB	78.88	68.26	77.84	72.6
Average	93.23	89.38	92.92	90.03

Table 12: Aurora 2 word accuracy using PSMAP for Set B

SNR (dB)	**Set B**			
	Restaurant	Street	Airport	Station
∞	99.6	99.6	99.6	99.6
20 dB	98.46	98.37	98.93	98.83
15 dB	97.18	97.58	97.7	97.96
10 dB	93.92	94.89	95.76	96.45
5 dB	87.14	89.69	90.58	91.21
0 dB	69.27	74.67	78.08	75.69
Average	89.19	91.04	92.21	92.03

Table 13: Aurora 2 word accuracy using PSMAP for Set C

SNR (dB)	**Set C**	
	Subway M	Street M
∞	99.6	99.6
20 dB	98.7	98.58
15 dB	98.4	97.67
10 dB	96.1	95.44
5 dB	91.46	89.18
0 dB	79.89	71.8
Average	92.92	90.53

significant because both LSI and PSMAP explicitly model the scatter of data around the mode (Equation (65)), therefore are expected to have similar accuracies. There are two principal reasons why LSI does not show improvements similar to PSMAP: (1) LSI uses a first order approximation to linearize the mode, and (2) the LSI system

uses the mode of static mel-cepstra to linearize the dynamic mel-cepstra coefficient.

Table 14: Aurora 2 average word accuracy comparisons of VTS, LSI, and PSMAP for Set A

SNR (dB)	**Set A**		
	VTS	LSI	PSMAP
∞	99.60	99.60	99.6
20	98.69	98.79	98.81
15	97.17	97.57	97.92
10	93.29	94.29	95.55
5	84.71	86.21	90.27
0	66.71	69.11	74.39
Average	88.11	89.19	91.39

Table 15: Aurora 2 average word accuracy comparisons of VTS, LSI, and PSMAP for Set B

SNR (dB)	**Set B**		
	VTS	LSI	PSMAP
∞	99.60	99.60	99.60
20	98.65	98.65	98.76
15	97.14	97.44	97.64
10	93.59	94.39	95.44
5	84.95	86.05	89.96
0	67.32	68.92	74.75
Average	88.33	89.09	91.31

Table 16: Aurora 2 average word accuracy comparisons of VTS, LSI, and PSMAP for Set C

SNR (dB)	**Set C**		
	VTS	LSI	PSMAP
∞	99.60	99.60	99.60
20	98.73	98.73	98.66
15	97.66	97.86	98.03
10	93.90	95.00	95.77
5	84.65	85.65	90.32
0	66.97	67.97	75.84
Average	88.38	89.04	91.72

Table 17 compares the average accuracy of various adaptation schemes to PSMAP-based adaptation scheme. The methods with qualifier 'NR' iteratively reestimate noise statistics using the adapted acoustic models. A single iteration for NR system consists of noise reestimation using the adapted model, followed by an update of the acoustic model itself. Three iterations of noise reestimation are performed for both VTS and LSI systems. The noise reestimated VTS and LSI schemes perform better than standard VTS and LSI systems.

Table 17: Aurora 2 average word accuracy comparisons for various model adaptation schemes

Algorithm	**Average Word Recognition Accuracy**
Baseline	63.38
CMN [6, Ch. 33]	70.88
CMVN [6, Ch. 33]	84.97
MLLR [6, Ch. 33]	76.6
VTS	88.27
VTS (NR) [74]	90.2
LSI [74]	89.11
LSI (NR)[74]	91.0
Multi-Style Models [6, Ch. 33]	90.06
PSMAP	**91.35**
ETSI-AFE(FE) [26]	87.02
CPSMAP-AFE(FE) (Tables 6, 7 and 8)	88.68

The systems with qualifier (FE) are front-end speech enhancement schemes. Both the systems are based on ETSI-AFE [26]. The comparison of front-end and back-end schemes is not fair but we have included these methods in Table 17 to provide the complete picture of enhancement schemes for robust ASR. It is worth noting that the CPSMAP-AFE (88.68%) has slightly better word accuracy than a standard VTS system (88.27%). It is important to note that a standard CPSMAP-based front-end has lower computational requirements than the VTS-based back-end.

The PSMAP-based acoustic model adaptation scheme outperforms all the other schemes including the VTS and LSI schemes with noise reestimation. It is unfair to

compare VTS to PSMAP directly, because PSMAP-based system uses stereo data (noisy speech and clean speech) during training. This added information provides PSMAP an advantage over VTS. It is more appropriate to compare PSMAP to LSI as both these algorithms are data driven and uses the same stereo data for training the models or splines. The other interesting fact to note is that even with noise reestimation LSI (91.0%) cannot outperform the PSMAP-based adaption system (91.42%).

5.5 Conclusions

Probabilistic space maps is a flexible framework that can be used map and extract relationships between data. In this chapter PSMAPs were successfully applied to acoustic model adaptation to mitigate the effect of noise on speech recognition. As discussed in the results section this adaptation scheme works well under various noise conditions and outperforms the standard adaptation algorithms. The PSMAP-based adaptation scheme outperforms traditional VTS/LSI schemes. We believe that PSMAP-based adaptation scheme will benefit from iterative noise reestimation. In the future it is our aim to integrate noise reestimation within the PSMAP framework.

Algorithms such as VTS, LSI, etc. only work for features such as MFCCs because the relationship among MFCCs of noisy speech, clean speech, and noise are known. These algorithms fail for features (e.g., HLDA) where relationship between clean speech and noisy speech is either unknown or difficult to model. PSMAP-based adaptation schemes do not require explicit knowledge of this relationship: given training data, PSMAPs can extract the mapping between features of clean and noisy speech. The PSMAP-based systems are also computationally more efficient than the VTS family of adaptation schemes.

CHAPTER VI

CONCLUSION AND FUTURE WORK

In this thesis we presented a probabilistic model of speech production. Unlike the speech inversion models, this new model does not aim to estimate the true vocal tract parameters. We start with the premise that the true VT configuration is hidden, and one can only make valid indirect observations about its orientation as a result of this premise, we model speech production using latent variables that represent true VT orientations. We call this graphical framework PSMAP.

PSMAPs do not invert the speech signal to extract articulatory information. This enables us to train PSMAPs using only speech data. PSMAPs capture the many-to-one mapping between the VTAF and speech spectra using a discrete probability matrix. Since PSMAPs do not use articulatory parameterization as observations, adaptation and application of PSMAPs to traditional speech enhancement algorithms is straightforward.

True articulator information boosts the performance of speech synthesizers and ASR. However, use of speech inversion algorithm in conjunction with speech enhancement or ASR systems almost never results in boosting the performance of later. This not the case with PSMAP-based enhancement or recognition systems as we have demonstrated through out this thesis. We verified our claims about PSMAP-based model of speech production using three speech applications: artificial bandwidth expansion, speech enhancement/noise suppression, and acoustic model adaptation.

The PSMAP based ABE system surpassed the state-of-the-art ABE system in quality. The proposed ABE system was able to synthesize artifact free broadband speech. The subjective and objective quality synthesized broadband speech was rated

consistently better than the quality of original narrowband speech.

Traditional noise suppression systems either improve perceptual quality of speech or improve the intelligibility of utterance. For the second application of PSMAPs we proposed a noise suppression system based on CPSMAPs that improved both. The speech synthesized by CPSMAP-based system had better subjective and objective quality than that of the speech synthesized by other standard speech enhancement systems. For Aurora 2, CPSMAP (88.68%) enhanced utterances had better recognition accuracy when compared to those enhanced with an ETSI-AFE (87.02%). For Aurora 2, the CPSMAP-based ASR (88.68%) system had better word accuracy than a standard VTS-based ASR (88.27%) system.

Acoustic model adaptation was the final application of PSMAP presented in this thesis. ASR systems achieve more benefit if we perform noise robust processing at the back-end (on the acoustic model instead of the features/utterance). PSMAP-based acoustic model adaptation scheme surpass most of the modern model adaptation schemes (including VTS and LSI with noise reestimation). The PSMAP based system is the first step; in the future we intend to add iterative noise reestimation to the PSMAP-based ASR back-end.

In closing, we want to highlight that PSMAP is a good model for speech production, it has versatile applications both on the perception and the recognition end of speech tasks. This model of speech production is easy to learn and train, as it does not require access to auxiliary articulatory information.

6.1 Summary of Contributions

Probabilistic Model of Speech Production:

- Using the equivalence between the inverse filters and uniform lossless tube models, we proposed and tested a probabilistic model of speech production that uses latent variables to model the acoustic articulator.

- We proposed and tested a graphical model, called PSMAPs, that captures the many-to-one mapping between the vocal tract area functions and speech spectra using latent states.
- We proposed and tested computationally efficient algorithms to train the PSMAPs and suggested a framework to impose sparsity on the mapping between the latent states that model vocal tract and speech spectra.
- We also proposed and tested improvements to simple/sparse PSMAPs to accommodate talker speed and vocal tract elasticity constraints. The anatomical constraints on the VT were used to impose temporal constraints on transition between the latent states of the subspace of PSMAPs.
- We presented efficient algorithms to train constraint-PSMAPs.
- We presented computationally efficient algorithms to perform inference on simple and constraint-PSMAPs.

Applications:

- We proposed and tested an algorithm based on PSMAP to perform artifical bandwidth expansion.
- We proposed and tested an artifact generation system to produce quantifiable, reproducible distortion in clean speech. This artifact generator provides baseline utterances to quantify and judge the quality of synthesized audio.
- We proposed and tested CPSMAP-based algorithm for estimation of *a priori* SNR.
- We proposed and tested a CPSMAP-based speech enhancement system.
- We proposed and tested CPSMAP-based mel-warped Wiener filter to enhance speech for the ASR system.

- We proposed and tested a PSMAP-based robust acoustic model adaptation system.

6.2 Future Work

Two avenues were explored in this thesis. First, we presented a model for speech production, and second we applied the new model to improve the performance of speech enhancement algorithms such as artificial bandwidth expansion, noise suppression, and acoustic model adaptation.

In the future we intend to improve the PSMAP-based acoustic model adaptation system by adding iterative noise reestimation to the existing setup. We believe that noise reestimation within the PSMAP framework will definitely improve the performance of the ASR system.

A future application of PSMAPs would be to the task of speaker separation. The idea would be to train separate PSMAPs for each speaker, and then use PSMAP trained for speaker 'X' to generate a mask that will be used to suppress 'X' from the mixture.

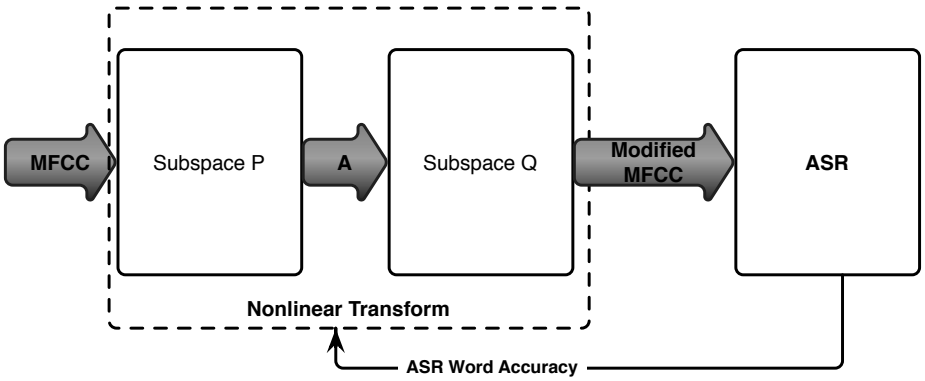


Figure 24: PSMAP as nonlinear transform.

Another future application will be to use PSMAP as nonlinear transforms that can be applied to acoustic features before they are passed to a recognizer. Figure 24 shows the block diagram for such a system.

A PSMAP in this setup will learn to counter system nonlinearities (e.g., microphone, channel, echo). PSMAPs used in this setup will be trained to optimize word error rates of the ASR.

APPENDIX A

USING LAMBERT $\mathcal{W}$ TO ESTIMATE $p(\rho_n)$ AND $p(\gamma_m|\rho_n)$

A Lambert $\mathcal{W}$ is a function of the form that satisfies following Equation (72):

$$\begin{aligned} \mathcal{W}(y)e^{\mathcal{W}(y)} &= y \\ \log \mathcal{W}(y) + \mathcal{W}(y) &= \log(y) \end{aligned} \tag{72}$$

Setting $y = e^x$ and in Equation (72) as follows

$$\begin{aligned} 0 &= -\mathcal{W}(e^x) - \log(\mathcal{W}(e^x)) + x \\ &= \frac{-1}{1/\mathcal{W}(e^x)} - \log(\mathcal{W}(e^x)) + x + \log(r) - \log(r) \\ &= \frac{-r}{r/\mathcal{W}(e^x)} + \log(r/\mathcal{W}(e^x)) + x - \log(r) \end{aligned} \tag{73}$$

Setting $x = 1 + \tau/\delta + \log(r)$ and $r = -\omega_n/p(\rho_n)$ Equation (73) becomes :

$$\begin{aligned} 0 &= \frac{\omega_n/\delta}{-(\omega_n/\delta)/\mathcal{W}(-\omega_n e^{1+\tau/\delta}/\delta)} + \log \frac{\omega_n/\delta}{\mathcal{W}(-\omega_n e^{1+\tau/\delta}/\delta)} \\ &\quad +1 + \frac{\tau}{\delta} \\ &= \frac{\omega_n/\delta}{p(\rho_n)} + \log p(\rho_n) + 1 + \frac{\tau}{\delta} \end{aligned} \tag{74}$$

Upon rearranging the terms of Equation (74) we get:

$$p(\rho_n) = \frac{-\omega_n/\delta}{\mathcal{W}(-\omega_n e^{1+\tau/\delta}/\delta)} \tag{75}$$

APPENDIX B

PSMAP TRAINING STRATEGIES

The EM algorithm is very sensitive to initialization, therefore this section will present some strategies to initialize and train PSMAPs to produce good subspace models. Initializing and training PSMAPs with a large number of bases is not a good idea. A better approach for creating large models is to initially started with a small models and gradually increase the size of the subspaces to reach a target.

Each subspace is initialized with global means and variance of data in that subspace. During the training it is possible that the Gaussian assigned to a latent state might degenerate, if the variance fall below a threshold, to prevent defunct Gaussians, variance of these Gaussians was floored with a predetermined variance. In our experiments we found that 2% of the global variance for the subspace was a good threshold.

The size subspace is increased by splitting the Gaussian with the maximum weight (prior probability $(p(\rho), p(\gamma))$), which are computed during training. Each Gaussian is split into two Gaussians. The variance of each child is same as that of the parent, but the means of children are perturbed by the $\pm 20\%$ of the standard deviation of each parent.

Updating the transition matrix $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ is a bit complicated and best if explained with an example. Consider a PSMAP with $N = 3$ hidden states in Subspace $\mathcal{P}$ and $M = 2$ hidden states in Subspace $\mathcal{Q}$ and transition matrix as shown by Equation (76). The size of both the Subspace $\mathcal{P}$ and Subspace $\mathcal{Q}$ is to be incremented by one. Based on the prior probabilities states $\rho = 2$ and $\gamma = 1$ in Subspaces $\mathcal{P}$ and Subspace $\mathcal{Q}$ were selected as candidates for the split. Splitting the Subspace $\mathcal{P}$

generates two new states, each of states will have the same mapping probabilities to states of Subspace $\mathcal{Q}$ as the parent state, algorithmically this can be achieved by replicating the columns of $\boldsymbol{A}$ for $\rho = 2$ and assigning it to the newly added states, see $\boldsymbol{A}_{\rho+}$(Equation (77)). When the size of the Subspace $\mathcal{Q}$ is incremented the probabilities of parent are split in half and assigned to each child, as seen from matrix $\boldsymbol{A}_{\rho+\gamma+}$(Equation (78)).

$$\boldsymbol{A} = \begin{bmatrix} 0.7 & 0.4 & 0.5 \\ 0.3 & 0.6 & 0.5 \end{bmatrix} \tag{76}$$

$$\boldsymbol{A}_{\rho+} = \begin{bmatrix} 0.7 & \mathbf{0.4} & 0.5 & \mathbf{0.4} \\ 0.3 & \mathbf{0.6} & 0.5 & \mathbf{0.6} \end{bmatrix} \tag{77}$$

$$\boldsymbol{A}_{\rho+\gamma+} = \begin{bmatrix} \mathbf{0.35} & \mathbf{0.2} & \mathbf{0.25} & \mathbf{0.2} \\ 0.3 & 0.6 & 0.5 & 0.6 \\ \mathbf{0.35} & \mathbf{0.2} & \mathbf{0.25} & \mathbf{0.2} \end{bmatrix} \tag{78}$$

The complete training strategy for PSMAPs is shown in Algorithm 4.

Algorithm 4 Complete algorithm for training a PSMAP.

1: Initialize PSMAP with global means and variances.
2: Initialize variance threshold for the subspaces.
3: **while** Model Size < TARGET SIZE **do**
4: Increment Model Size (Split the states of subspaces).
5: Train the model generated in Step 4 using the method described in Section 2.3.
6: **end while**

APPENDIX C

AFE: SPEECH ENHANCEMENT SYSTEM

Figure 25 shows the block diagram of the ETSI-AFE based noise suppression system. The details about each block can be found in the standards document [26]. In the next section we describe the algorithm used in the voice activity detection (VAD) block. The algorithm used for VAD in this thesis is a slightly different than the one used in ETSI-AFE.

For comparisons of various noise suppression systems we only replace the Wiener filter (WF) design block. A simple DD Wiener filter is used for the standard AFE system. The WF block is replaced by a CPSMAP *a priori* SNR estimator to build a CPSMAP-AFE noise suppression system.

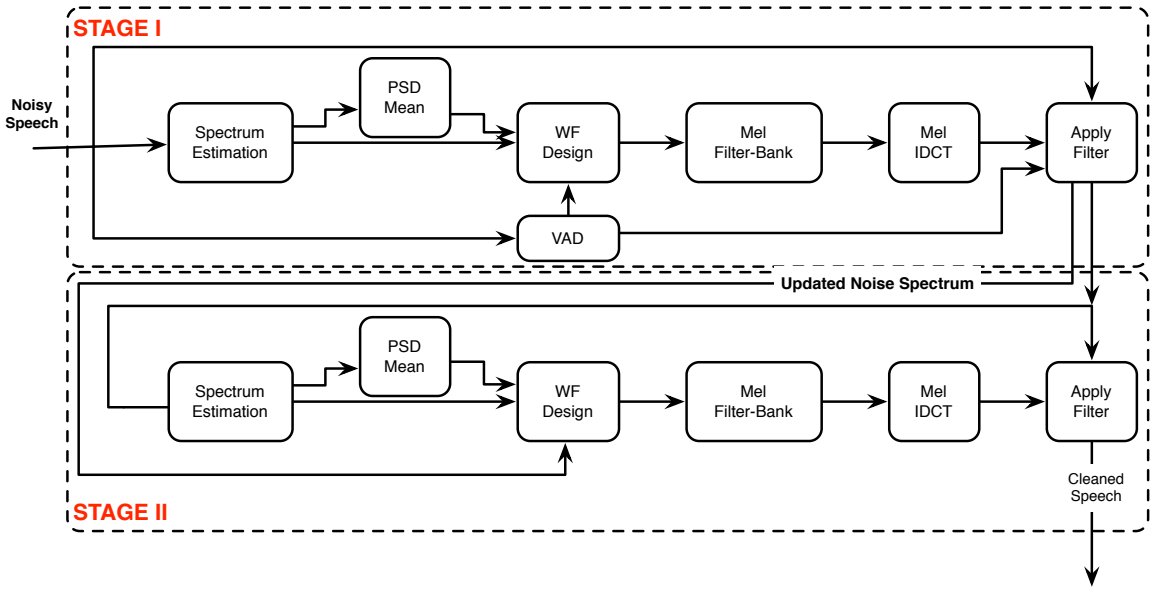


Figure 25: Block diagram of ETSI-AFE speech enhancement system

C.1 Algorithm for Voice Activity Detection

```
% Compute the gain for the exponential window


% FRAME_THRESHOLD = 15
% LAMBDA_LTE = 0.97
if(t < FRAME_THRESHOLD)
    lambdaLTE = 1 - 1/t
else
    lambdaLTE = LAMBDA_LTE
end


% Get the frame energy
frameEn = 0.5 + 16/ln(2)*(ln(1 + frame*frame'/64))


%  Update the mean Energy value


% SNR_THRESHOLD_LTE = 20
% MIN_FRAME = 10
% ENERGY_FLOOR = 100
% lambdaLTEhigh = 0.99


if((frameEn - meanEn) < SNR_THRESHOLD_LTE  or t < MIN_FRAME)
    if(frameEn < meanEn or t < MIN_FRAME)
        meanEn = meanEn + (1 - lambdaLTE)*(frameEn - meanEn)
    else
        meanEn = meanEn + (1 - lambdaLTEhigh)*(frameEn - meanEn)
    end
```

```
    if(meanEn < ENERGY_FLOOR)
        meanEn = ENERGY_FLOOR
    end
end

% The value of frame energy (frameEn) and meanEn is used to make a VAD
% decision speech (flagVAD = 1) or nonspeech (flagVAD = 0)

% SNR_THRESHOLD_VAD = 15
% MIN_SPEECH_FRAME_HANGOVER = 4
% HANGOVER = 15

if(t > 4)
    if(frameEn - meanEn) > SNR_THRESHOLD_VAD)
        flagVAD = 1 % SPEECH
        nbSpeechFrame = nbSpeechFrame + 1
    else
        if( nbSpeechFrame >  MIN_SPEECH_FRAME_HANGOVER)
            hangOver = HANGOVER
            nbsSpeechFrame = 0
        end
        if(hangOver != 0)
            hangOver = hangOver - 1
            flagVAD = 1
        else
            flagVAD = 0
```

```
        end
    end
end

% flagVAD, meanEn and nbSpeechFrame are initlized to zero. The frame index
% t is initilized to 0 and incremented by 1 for every frame processed
```

REFERENCES

[1] "ITU-T P.800. Methods for subjective determination of transmission quality - Series P: telephone transmission quality; methods for objective and subjective assessment of quality," Aug. 1996.

[2] Anton, H., Bivens, I., and Davis, S., *Calculus*. Wiley, 2001.

[3] Atal, B. S., Chang, J. J., Mathews, M. V., and Tukey, J. W., "Inversion of articulatorytoacoustic transformation in the vocal tract by a computersorting technique," *Journal of the Acoustical Society of America*, vol. 63, no. 5, pp. 1535–1555, 1978.

[4] Atal, B. S. and Hanauer, S. L., "Speech analysis and synthesis by linear prediction of speech wave," *The Journal of Acoustical Society of America*, vol. 1, pp. 637–655, August 1971.

[5] Avendano, C., Hermansky, H., and Wan, E., "Beyond Nyquist: Towards the recovery of broad-bandwidth speech from narrow-bandwidth speech," *Eurospeech*, pp. 165–168, 1995.

[6] Benesty, J., Sondhi, M. M., and Huang, Y. A., *Springer Handbook of Speech Processing*, ch. 33. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.

[7] Bishop, C. M., *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, August 2006.

[8] Boll, S., "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, pp. 113–120, Apr 1979.

[9] Brand, M., "Pattern discovery via entropy minimization," *Proc. Artificial Intelligence and Statistics*, 1998.

[10] Chen, B. and Loizou, P., "A laplacian-based mmse estimator for speech enhancement," *Speech Communication*, vol. 49, pp. 134–143, 2007.

[11] Chen, B. and Loizou, P., "Speech enhancement using a mmse short time spectral amplitude estimator with laplacian speech modeling," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on*, vol. 1, pp. 1097 – 1100, 18-23 2005.

[12] Chennoukh, S.and Gerritis, A., Miet, G., and Sluijter, R., "Speech enhancement via frequency bandwidth extension using line spectral frequencies," *International Conference on Acoustics, Speech, and Signal Processing*, pp. 665 – 668, 2001.

[13] Cohen, I., "Speech enhancement using a noncausal a priori SNR estimator," *IEEE Signal Processing Letters*, vol. 11, pp. 725–728, 2004.

[14] Cohen, I., "Speech enhancement using super-gaussian speech models and noncausal a priori SNR estimation," *Speech Communication*, vol. 47, pp. 336–350, 2005.

[15] Corless, R. M., Gonnet, G. H., Hare, D. E. G., Jerey, D. J., and Knuth, D. E., "On the lambert w function," *Advances in Computational Mathematics*, pp. 329–359, 1996.

[16] Cover, T. and Thomas, J., *Elements of information theory*. New York: Wiley, 1991.

[17] Deller, J. R., Hansen, J. H. L., and Proakis, J. G., *Discrete-Time Processing of Speech Signals*. IEEE Press.

[18] Dempster, A., Laird, N., and Rubin, D., "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[19] Dempster, A. P., Laird, N. M., and Rubin, D. B., "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.

[20] Deng, L., Droppo, J., and Acero, A., "Enhancement of log mel power spectra of speech using a phase-sensitive model of the acoustic environment and sequential estimation of the corrupting noise," *IEEE Transactions on Speech and Audio Processing,*, vol. 12, pp. 133–143, March 2004.

[21] Digalakis, V., Rtischev, D., Neumeyer, L., and Sa, E., "Speaker adaptation using constrained estimation of gaussian mixtures," *IEEE Transactions on Speech and Audio Processing*, vol. 3, pp. 357–366, 1995.

[22] Ephraim, Y. and Malah, D., "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 32, pp. 1109–1121, Dec 1984.

[23] Ephraim, Y. and Malah, D., "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, pp. 443–445, 1985.

[24] Erkelens, J., Jensen, J., and Heusdens, R., "A general optimization procedure for spectral speech enhancement methods," EUSIPCO 2006 - Italy, 2006.

[25] ES 201 108 Ver. 1.1.3, *Speech Processing, Transmission and Quality Aspects (STQ); Distributed speech recognition; Front-end feature extraction algorithm; Compression algorithms.*

[26] ETSI ES 201 050 Ver 1.1.5, *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced Front-End Feature Extraction Algorithms; Compression Algorithms.*

[27] Fingscheidt, T., "Data-driven speech enhancement," Sprachkommunikation 2006 - ITG-Fachtagung, Apr 2006.

[28] Fingscheidt, T., Suhadi, S., and Stan, S., "Environment-optimized speech enhancement," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 825 –834, may 2008.

[29] Flanagan, J. L., *Speech analysis synthesis and perception.*

[30] Frankel, J., Richmond, K., King, S., and Taylor, P., "An automatic speech recognition system using neural networks and linear dynamic models to recover and model articulatory traces," in *International Conference on Spoken Language Processing (ICSLP)*, 2000.

[31] Frankel, J. and Simon, K., "Asr - articulatory speech recognition," in *Proceedings of Eurospeech 2001*, 2001.

[32] Gales, M. J. F., "Maximum likelihood linear transformations for hmm-based speech recognition," *Computer Speech and Language*, vol. 12, pp. 75–98, 1998.

[33] Gales, M. J. F. and Young, S. J., "Robust speech recognition in additive and convolutional noise using parallel model combination," *Computer Speech and Language*, vol. 9, pp. 289–307, 1995.

[34] Gales, M. and Woodland, P., "Mean and variance adaptation within the MLLR framework," *Computer Speech and Language*, vol. 10, pp. 249–264, 1996.

[35] Gauvain, J. and Lee, C.-H., "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and Audio Processing*, vol. 2, pp. 291–298, 1994.

[36] Gong, Y., "Speech recognition in noisy environments: A survey," *Speech Communication*, vol. 16, no. 3, pp. 261 – 291, 1995.

[37] Gopinath, R. A., Gales, M. J., Gopalakrishnan, P. S., and Picheny, M. A., "Robust speech recognition in noise performance of the ibm continuous speech recognizer on the arpa noise spoke task," in *ARPA Workshop on Spoken Language System Technology*, pp. 127–130, 1995.

[38] Hasan, M., Salahuddin, S., and Khan, M., "A modified a priori snr for speech enhancement using spectral subtraction rules," *Signal Processing Letters, IEEE*, vol. 11, pp. 450 – 453, april 2004.

[39] Hirsch, H. and Pearce, D., "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proceedings of ASR*, 2000.

[40] Hosoki, M., Nagai, T., and Kurematsu, A., "Speech signal bandwidth extension and noise removal using subband hmm," *International Conference on Acoustics, Speech, and Signal Processing*, pp. 245–248, 2002.

[41] Hoyer, P. O., "Non-negative matrix factorization with sparseness constraints," *Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[42] Hu, Y. and Huo, Q., "An HMM compensation approach using unscented transformation for noisy speech recognition," in *International Conference on Spoken Language Processing*, pp. 346–357, 2006.

[43] Iskarous, K., Goldstein, L., Whalen, D., Tiede, M., and Rubin, E., "Casy: The haskins configurable articulatory synthesizer," in *International Congress of Phonetic Sciences, Barcelona Spain*, 2003.

[44] ITU-T, "Perceptual evaluation of speech quality (pesq): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs," Tech. Rep. P.862-02, ITU-T, 2001.

[45] Jax, P. and Vary, P., "Artificial bandwidth extension of speech signals using MMSE estimation based on a hidden Markov model," *International Conference on Acoustics, Speech, and Signal Processing*, no. 680-683, 2003.

[46] Julier, S. J. and Uhlmann, J. K., "Unscented filtering and nonlinear estimation," *Proceedings of the IEEE*, vol. 92, no. 3, pp. 401–422, 2004.

[47] Kalgaonkar, K. and Clements, M. A., "Vocal tract and area function estimation with both lip and glottal losses," in *Interspeech, Antwerp Belgium*, pp. 550–553, 2007.

[48] Kalgaonkar, K. and Clements, M. A., "Vocal tract area based artificial bandwidth extension,," *IEEE Workshop on Machine Learning for Signal Processing, Cancun, Mexico*, 2008.

[49] Kalgaonkar, K. and Clements, M. A., "Constrained probabilistic subspace maps applied to speech enhancement," in *Interspeech London UK*, 2009.

[50] Kalgaonkar, K. and Clements, M. A., "Sparse probabilistic space mapping and it application to speech bandwidth expansion," in *IEEE International Conference on Acoustics Speech and Signal Processing*, 2009.

[51] Kalgaonkar, K. and Clements, M., "Hmm adaptation using sparse probabilistic space mapping for noisy speech," in *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 4554 –4557, march 2010.

[52] Kalgaonkar, K., Seltzer, M., and Acero, A., "Noise robust model adaptation using linear noise robust model adaptation using linear spline interpolation," in *IEEE workshop on Automatic Speech Recognition and Understanding*, December 2009.

[53] Kobayashi, T., Yagyu, M., and Shirai, K., "Application of neural networks to articulatory motion estimation," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 489–492, 1991.

[54] Leggetter, C. J. and Woodland, P. C., "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.

[55] Li, J., Deng, L., Yu, D., Gong, Y., and Acero, A., "High-performance hmm adaptation with joint compensation of additive and convolutive distortions via vector taylor series," in *IEEE Workshop on Automatic Speech Recognition Understanding*, pp. 65–70, 2007.

[56] Lu, Y. and yang Wu, Z., "Model adaptation algorithm using vector taylor series," *Journal of Electronics Information and Technology*, vol. 2010, pp. 107–111, 2010.

[57] Maeda, S., *Speech Production and Speech Modeling*, pp. 131–149. Kluwer, 1990.

[58] Markel, J. D. and Gray, A. H., *Linear Prediction of Speech*. Springer-Verlag, 1976.

[59] Martin, R., "Speech enhancement using mmse short time spectral estimation with gamma distributed speech priors," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, may 2002.

[60] Martin, R. and Breithaupt, C., "Speech enhancement in the dft domain using laplacian speech priors," in *International Workshop on Acoustic Echo and Noise Control*, 2003.

[61] McAulay, R. and Malpass, M., "Speech enhancement using a soft-decision noise suppression filter," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, pp. 137–145, Apr 1980.

[62] Moreno, P., Raj, B., and Stern, R., "A vector taylor series approach for environment-independent speech recognition," in *IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 2, pp. 733–736 vol. 2, May 1996.

[63] Neiberg, D., Ananthakrishnan, G., Engwall, O., and Mocha-ema, B. D., "The acoustic to articulation mapping: Non-linear or non-unique?," in *Interspeech*, 2008.

[64] Okadome, T., Suzuki, S., and Honda, M., "Recovery of articulatory movements from acoustics with phonemic information," in *Proc. 5th Seminar on Speech Production.*, 2000.

[65] Oppenheim, A. V., Schafer, R. W., and Buck, J. R., *Discrete-time signal processing (2nd ed.)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1999.

[66] Paul, D. and Baker, J., "The design for the wall street journal-based csr corpus," in *International Conference on Spoken Language Processing*, 1992.

[67] Porter, J. and Boll, S., "Optimal estimators for spectral restoration of noisy speech," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 9, pp. 53 – 56, mar 1984.

[68] Quatieri, T. F., *Discrete-Time Speech Signal Processing: Principles and Practice*. Prentice Hall, Nov.

[69] Rabiner, L. and Juang, B., "An introduction to hidden markov models," *ASSP Magazine, IEEE*, vol. 3, pp. 4 – 16, jan 1986.

[70] Rahim, M. G., Goodyear, C. C., Kleijn, W. B., Schroeter, J., and Sondhi, M. M., "On the use of neural networks in articulatory speech synthesis," *Journal of the Acoustical Society of America*, vol. 93, no. 2, 1993.

[71] Riedmiller, M. and Braun, H., "A direct adaptive method for faster backpropagation learning: The rprop algorithm," in *IEEE International Conference on Neural Networks*, pp. 586–591, 1993.

[72] Rothenberg, M., "A multichannel electroglottograph," *Journal of Voice*, vol. 6, pp. 36–43, 1992.

[73] Scanlon, M., "Acoustic sensor for health status monitoring," *Proceedings of IRIS Acoustic and Seismic Sensing*, vol. 2, pp. 205–222, 1998.

[74] Seltzer, M., Acero, A., and Kalgaonkar, K., "Acoustic model adaptation via linear spline interpolation for robust speech recognition," in *IEEE International Conference on Acoustics Speech and Signal Processing*, pp. 4550 –4553, march 2010.

[75] Shirai, K. and Kobayashi, T., "Estimating articulatory motion from speech wave," *Speech Communication*, vol. 5, pp. 159–170, 1986.

[76] Sondhi, M. M., "Estimation of vocal tract areas: the need for acoustical measurments," *IEEE Transactions on Acoustic Speech and Signal Processing*, vol. 27(3), pp. 268–273, June 1979.

[77] Sondhi, M. and Schroeter, J., "A hybrid time-frequency domain articulatory speech synthesizer," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, pp. 955–967, 1987.

[78] Stevens, K. N., *Acoustic Phonetics*. Cambridge, MA: MIT Press, 1998.

[79] Suhadi, S., Last, C., and Fingscheidt, T., "A data-driven approach to a priori snr estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, pp. 186 –195, jan. 2011.

[80] Teacher, C. and Coulter, D., "Performance of lpc vocoders in a noisy environment," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 216 – 219, 1979.

[81] Varga, A., Steeneken, H. J. M., Tomlinson, M., and Jones, D., "The noisex-92 study on the effect of additive noise on automatic speech recognition," 1992.

[82] Voran, S., "Listener ratings of speech passbands," *IEEE workshop on Speech Coding*, pp. 81–82, 1997.

[83] Wakita, H., "Direct estimation of the vocal tract shape by inverse filtering of acoustic speech waveform," *IEEE Transactions on Audio and Electroacoustics*, vol. 21(5), pp. 417–427, October 1973.

[84] Wang, D. and Lim, J., "The unimportance of phase in speech enhancement," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 30, pp. 679 – 681, aug 1982.

[85] Wolfe, P. J. and Godsill, S. J., "Simple alternatives to the ephraim and malah suppression rule for speech enhancement," in *Proceedings of the IEEE Signal Processing Workshop on Statistical Signal Processing*, pp. 496 – 499, 2001.

[86] Wrench, A., "A new resource for production modeling in speech technology," in *Workshop on Innovations in Speech Processing UK*, 2001.

[87] Yasukawa, W., "Signal restoration of broadband speech using nonlinera processing," *Proc. European Signal Processing Conference*, no. 176-178, 1996.

[88] Yoshida, Y. and Abe, M., "An algorithm to reconstruct wideband speech from narrowband speech based on codebook mapping," *International Conference on Spoken Language Processing*, pp. 1591–1594, 94.

[89] Young, S., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., and Woodland, P., *The HTK Book Version 3.0*. Cambridge University Press, 2000.

VITA

Kaustubh Kalgaonkar was born in M.P., India. He received his Bachelor of Engineering (B.E.) from University of Pune, Maharashtra, India in 2001. He graduated at the top of his class of 2001. He has earned his Masters of Science (M.S.) in 2004 at University of Missouri, Columbia. Since 2005 Kaustubh has been pursuing a Ph.D. degree at Georgia Institute of Technology in Atlanta, Georgia. He is a recipient of Outstanding Researcher award at the Center for Signal and Image Processing at Georgia Institute of Technology. Before his graduate studies he took sequence of short employments at Motorola and Sprint. During his graduate studies Kaustubh took a series of internships at Mitsubishi Electronics Research Laboratories (MERL) and Microsoft Research (MSR).

His research interests include speech enhancement, speech recognition. He is also interested in the use of non-acoustic auxiliary sensors such as ultrasonic sensor for speech enhancement and surveillance.

Kaustubh is a member of Tau Beta Pi, Sigma Xi, and IEEE.